



San Diego, California 92152

MARINE PHYSICAL **DTIC** FILE COPY  
LABORATORY  
SCRIPPS INSTITUTION OF OCEANOGRAPHY

4

PARABOLIC EQUATION MODEL

Jean-Marie Tran

MPL TECHNICAL MEMORANDUM 401

MPL-U-9/88

*Approved for public release; distribution unlimited.*

April 1988

**DTIC**  
**ELECTE**  
**JUL 06 1988**  
**S H D**

AD-A197 480

## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>			2. RESTRICTIVE MARKINGS		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION AVAILABILITY OF REPORT		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE			Approved for public release; distribution unlimited.		
4. PERFORMING ORGANIZATION REPORT NUMBER(S) <b>MPL Technical Memorandum 401 (FPL-U-9/88)</b>			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION <b>Marine Physical Laboratory</b>		6b. OFFICE SYMBOL (If applicable) <b>MPL</b>		7a. NAME OF MONITORING ORGANIZATION <b>Office of Naval Research Department of the Navy</b>	
6c. ADDRESS (City, State, and ZIP Code) <b>University of California, San Diego Scripps Institution of Oceanography San Diego, CA 92152</b>		7b. ADDRESS (City, State, and ZIP Code) <b>800 North Quincy Street Arlington, VA 22217-5000</b>			
8a. NAME OF FUNDING/SPONSORING ORGANIZATION <b>Office of Naval Research</b>		8b. OFFICE SYMBOL (If applicable) <b>ONR</b>		9. MONITORING ORGANIZATION IDENTIFICATION NUMBER <b>N00014-87-C-0127</b>	
8c. ADDRESS (City, State, and ZIP Code) <b>Department of the Navy 800 North Quincy Street Arlington, VA 22217-5000</b>		10. SOURCE OF FUNDING NUMBERS			
		PROGRAM ELEMENT NO		PROJECT NO	
				TASK NO	
				WORK UNIT ACCESSION NO	
11. TITLE (Include Security Classification) <b>PARABOLIC EQUATION MODEL</b>					
12. PERSONAL AUTHOR(S) <b>Jean-Marie Tran</b>					
13a. TYPE OF REPORT <b>Summary</b>		13b. TIME COVERED FROM TO		14. DATE OF REPORT (Give Month, Day) <b>April 1988</b>	
				15. PAGE COUNT <b>58</b>	
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB-GROUP			
			parabolic propagation model, acoustic modelling, SACLANT PAREQ program		
19. ABSTRACT					
<p>This report deals with a parabolic propagation model now implemented at the Marine Physical Laboratory. This parabolic equation (PE) program is a FORTRAN 77 version of the SACLANT PAREQ program [1].</p> <p>The parabolic methods were a major development in the field of acoustic modeling [2]. The advantage of the method is the simple way range dependence is handled. Diffraction effects are included, but neither reverberation nor backscattering are accounted for [3]. The parabolic equation requires small angles of propagation from the horizontal and cannot handle steep sound velocity gradients or high frequency propagation because of computer speed considerations. The running time is proportional to <math>Hf^{2.5} \frac{dc}{dz} R_{max}</math> where <math>H</math> is the water plus bottom height, <math>f</math> the frequency, <math>c</math> the sound velocity, <math>z</math> the depth and <math>R_{max}</math> the maximum range [4].</p> <p>In the following pages, the parabolic equations are derived, the associated numerical algorithms are presented, the environment and the program inputs are described. Finally examples are given.</p> <p>The goal of this report is to provide the potential user the necessary information to understand and use the PE program.</p>					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT <input type="checkbox"/> UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS			21. ABSTRACT SECURITY CLASSIFICATION <b>UNCLASSIFIED</b>		
22a. NAME OF RESPONSIBLE INDIVIDUAL <b>W.S. Hodgkiss</b>			22b. TELEPHONE (Include Area Code) <b>(619) 534-1798</b>		22c. OFFICE SYMBOL <b>MPL</b>

# PARABOLIC EQUATION MODEL

Jean-Marie Tran

Marine Physical Laboratory  
Scripps Institution of Oceanography  
San Diego, CA 92152

## ABSTRACT

This report deals with a parabolic propagation model now implemented at the Marine Physical Laboratory. This parabolic equation (PE) program is a FORTRAN 77 version of the SACLANT PAREQ program [1].

The parabolic methods were a major development in the field of acoustic modeling [2]. The advantage of the method is the simple way range dependence is handled. Diffraction effects are included, but neither reverberation nor backscattering are accounted for [3]. The parabolic equation requires small angles of propagation from the horizontal and cannot handle steep sound velocity gradients or high frequency propagation because of computer speed considerations. The running time is proportional to  $Hf^2 \left[ \frac{\partial c}{\partial z} \right]_{\max} R_{\max}$  where  $H$  is the water plus bottom height,  $f$  the frequency,  $c$  the sound velocity,  $z$  the depth and  $R_{\max}$  the maximum range [4].

In the following pages, the parabolic equations are derived, the associated numerical algorithms are presented, the environment and the program inputs are described. Finally examples are given.

The goal of this report is to provide the potential user the necessary information to understand and use the PE program.



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	

## **CONTENTS**

- 1. Parabolic Approximation**
- 2. Parabolic Equations**
- 3. Numerical Implementation**
- 4. Interpretation of the Split Step Algorithm**
- 5. Description of MPL's PE Program**
- 6. Phase Error Corrections**
- 7. Source Field**
- 8. Description and Treatment of the Environment**
- 9. Program Inputs**
- 10. Program Output**
- 11. Examples**
- 12. References**

## 1. Parabolic Approximation

The starting point of acoustic models is the homogeneous Helmholtz equation or reduced wave equation which is an elliptic partial differential equation :

$$\nabla^2 p + k^2 p = 0 \quad (1.1)$$

$$\nabla^2 p + k_o^2 n^2 p = 0 \quad (1.2)$$

with  $k^2 = k_o^2 n^2$  where  $k_o$  is a reference wave number and  $n$  can be regarded as a refraction index. In cylindrical coordinates, neglecting the azimuthal dependence, the Helmholtz equation is :

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial z^2} + k_o^2 n^2 p = 0 \quad (1.3)$$

Intrinsic cylindrical spreading is accounted for by a factor  $r^{-1/2}$  in the pressure, leading to the classic change of variable  $p = r^{-1/2} \Phi(r, z)$ .

Then, the equation becomes :

$$\frac{\partial^2 \Phi}{\partial r^2} + \frac{\partial^2 \Phi}{\partial z^2} + (k_o^2 n^2 + \frac{1}{4r^2}) \Phi = 0 \quad (1.4)$$

Essentially, the far field zone is of interest ; therefore,  $k_o r \gg 1$  is assumed with an index of refraction of the order of unity. The equation is simplified

$$\frac{\partial^2 \Phi}{\partial r^2} + \frac{\partial^2 \Phi}{\partial z^2} + k_o^2 n^2 \Phi = 0 \quad (1.5)$$

or

$$(\frac{\partial}{\partial r^2} + k_o^2 Q^2) \Phi = 0 \quad (1.6)$$

where the operator  $Q$  is

$$Q^2 = n^2 + \frac{1}{k_o^2} \frac{\partial^2}{\partial z^2} \quad (1.7)$$

Parabolic approximations are obtained by factorization of the differential operator  $Q$  into the product of two commuting operators

$$\left(\frac{\partial^2}{\partial r^2} + k_o^2 Q^2\right) = \left(-i\frac{\partial}{\partial r} + k_o Q\right) \left(i\frac{\partial}{\partial r} + k_o Q\right) \quad (1.8)$$

It requires that  $\frac{\partial}{\partial r}$  and  $Q$  commute. This condition is considered satisfied if the index of refraction slowly varies in range so that the gradient  $\frac{\partial n}{\partial r}$  is negligible.

The differential equation then can be separated into two equations :

$$\left(i\frac{\partial}{\partial r} + k_o Q\right) \Phi = 0 \quad (1.9)$$

$$\left(-i\frac{\partial}{\partial r} + k_o Q\right) \Phi = 0 \quad (1.10)$$

The general solution is the sum of their two solutions, the first one represents the outgoing wave and the second one the incoming wave. The underlying nature of the parabolic approximation is to consider that the solution propagates primarily in the outward direction with very little reflection, backscattering and reverberation.

## 2. Parabolic Equations

The differences in formulations of the parabolic approximations leading to different parabolic equations reside in the way the  $Q$  operator is evaluated. The pseudo differential operator  $Q$  is rewritten as

$$Q = (1 + n^2 - 1 + \frac{1}{k_o^2} \frac{\partial^2}{\partial z^2})^{1/2} \quad (2.1)$$

and linearized in different ways by assuming either  $(n^2 - 1)\Phi$  or  $\frac{1}{k_o^2} \frac{\partial^2 \Phi}{\partial z^2}$  is small compared to  $\Phi$ .

Taking  $n^2 - 1 \ll 1$  is equivalent to considering small gradients of the sound velocity. Tappert [6] shows that the term  $\frac{1}{k_o^2} \frac{\partial^2 \Phi}{\partial z^2}$  is related to the mean square angle of propagation with respect to the horizontal. If  $\Phi = \exp ik_o(r \cos \theta \pm z \sin \theta)$ , the norm of  $\frac{1}{k_o^2} \frac{\partial^2 \Phi}{\partial z^2}$  is  $\sin^2 \theta$  where  $\theta$  is the angle of propagation with respect to the horizontal. Therefore assuming it is small is equivalent to considering small propagation angles with respect to the horizontal.

Approximations of the  $Q$  operator lead to two classes of models, one of iterative methods based on finite difference schemes, the other based on the Split Step Algorithm [6,7] using Fast Fourier Transforms (FFTs). Only the last class will be described here in relation to MPL's PE program.

The standard small angle parabolic approximation is obtained by assuming that both the index of refraction term and the propagation angle term are small and by using a first order Taylor series [5,6].

$$Q \approx \left[ 1 + \frac{n^2 - 1}{2} + \frac{1}{2k_o^2} \frac{\partial^2}{\partial z^2} \right] \quad (2.2)$$

The corresponding equation is obtained by the classic transformation  $\Phi = e^{ik_0 r} \Psi$  :

$$i \frac{\partial}{\partial r} (e^{ik_0 r} \Psi) + k_0 \left[ 1 + \frac{n^2 - 1}{2} + \frac{1}{2k_0} \frac{\partial^2}{\partial z^2} \right] (e^{ik_0 r} \Psi) = 0 \quad (2.3)$$

$$i \frac{\partial \Psi}{\partial r} + k_0 \frac{n^2 - 1}{2} \Psi + \frac{1}{2k_0} \frac{\partial^2 \Psi}{\partial z^2} = 0 \quad (2.4)$$

$$\frac{\partial^2 \Psi}{\partial z^2} + 2ik_0 \frac{\partial \Psi}{\partial r} + k_0^2 (n^2 - 1) \Psi = 0 \quad (2.5)$$

Equation (2.5) is the standard parabolic equation, also called Tappert and Hardin parabolic equation.

Thompson and Chapman [7] introduced an improved parabolic approximation by using another development of  $Q$  proposed by Feit and Fleck [8] :

$$Q = \sqrt{1 + \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2}} + \sqrt{1 + n^2 - 1} - 1 \quad (2.6)$$

$$Q = \sqrt{1 + \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2}} + n - 1 \quad (2.7)$$

They showed that this approximation can stand much wider angles. While the standard parabolic equation is valid to about  $\pm 20^\circ$ , this modified parabolic equation is valid up to  $\pm 40^\circ$  [4]. Using an additional pseudo-operator identity

$$(1 + \mu)^{\frac{1}{2}} = 1 + \frac{\mu}{(1 + \mu)^{\frac{1}{2}} + 1} \quad (2.8)$$

the equation becomes

$$i \frac{\partial \Phi}{\partial r} + k_0 \left[ 1 + \frac{\frac{1}{k_0^2} \frac{\partial^2}{\partial z^2}}{(1 + \frac{1}{k_0^2} \frac{\partial^2}{\partial z^2})^{\frac{1}{2}} + 1} + n - 1 \right] \Phi = 0 \quad (2.9)$$

With the previous envelope transformation  $\Phi = e^{ik_0 r} \Psi$ , one gets

$$i \frac{\partial \Psi}{\partial r} + k_0 \frac{\frac{1}{k_0^2} \frac{\partial^2 \Psi}{\partial z^2}}{(1 + \frac{1}{k_0^2} \frac{\partial^2 \Psi}{\partial z^2})^{\frac{1}{2}} + 1} + k_0 (n - 1) \Psi = 0 \quad (2.10)$$



### 3. Numerical Implementation

These parabolic equations are extensively used because of their efficient numerical implementation based on FFTs. Jensen and Krol [2,9] elegantly derive the Split Step algorithm for the Tappert and Hardin parabolic equation. They do its Fourier transform and assume an almost constant index of refraction. By inverse Fourier Transform, they get

$$\Psi = e^{i \frac{k_0}{2} (n^2 - 1) (r - r_0)} F^{-1} \left[ e^{i \frac{(r - r_0) s^2}{2k_0}} F(\Psi(r_0, z)) \right] \quad (3.1)$$

where  $F$  is the one dimensional Fourier transform with respect to  $z$ ,  $r_0$  is a reference range, and  $s$  is the Fourier variable.

The above algorithm was primarily derived by Tappert and Hardin [5,6] using operator representations. The parabolic differential equation is

$$\frac{\partial \Psi}{\partial r} = iA \Psi + iB \Psi \quad (3.2)$$

with  $A(r, z) = \frac{k_0}{2} (n^2 - 1)$  and  $B(z) = \frac{1}{2k_0} \frac{\partial^2}{\partial z^2}$ . Note that at this point, attenuation can

be introduced by inserting a complex term in  $A$  :  $A = \frac{k_0}{2} (n^2 - 1 + i\alpha)$ . The solution is then assumed to be

$$\Psi(r_0 + \Delta r, z) = \left[ \exp \left\{ i \int_{r_0}^{r_0 + \Delta r} A(r, z) dr + i \int_{r_0}^{r_0 + \Delta r} B(z) dr \right\} \right] \Psi(r_0, z) \quad (3.3)$$

by analogy with differential equations in the real space with constant coefficients  $A, B$ . Assuming that  $A$  varies slowly with respect to  $r$ , one splits the operator  $\Psi(r_0 + \Delta r, z) = \exp(i \Delta r A) \exp(i \Delta r B) \Psi(r_0, z)$ . The term  $\exp(i \Delta r B) \Psi(r_0, z)$  is evaluated via its Fourier transform

$$\exp(i \Delta r B) \Psi(r_0, z) = F^{-1} \left[ e^{-i \frac{\Delta r s^2}{2k_0}} F(\Psi(r_0, z)) \right] \quad (3.4)$$

One can easily derive the Split Step Algorithm for the modified parabolic equation (Thomson and Chapman) :

$$\Psi(r, z) = e^{i \frac{k_o}{2} (n-1) (r-r_o)} F^{-1} \left[ \exp \left( \frac{-i \frac{s^2}{2k_o} (r - r_o)}{(1 - \frac{s^2}{k_o^2})^{1/2} + 1} \right) F \left[ \Psi(r_o, z) \right] \right] \quad (3.5)$$

#### 4. Interpretation of the Split Step Algorithm

Two physical phenomena are involved : the propagation in a homogeneous environment and the influence of changes in the medium on the propagation. The Split Step Algorithm multiplies the angular spectrum by a phase shift to account for propagation over a range step  $\Delta r$ . The steeper the angle, the greater is the phase shift. Then it multiplies the inverse Fourier transform of this new range step spectrum by the index of refraction term which accounts for changes in the medium.

Other algorithms for the standard parabolic equation were introduced [6,10] such as

$$\Psi(r_0 + \Delta r, z) = \exp\left[\frac{i\Delta r k_0(n^2 - 1)}{4}\right] F^{-1}\left\{\exp\frac{i\Delta r k^2}{2k_0}\left[F\left(\exp\left[\frac{i\Delta r k_0(n^2 - 1)}{4}\right]\Psi(r_0, z)\right)\right]\right\} \quad (4.1)$$

It allows for a better coupling between the two physical phenomena described above. An error analysis [10] shows that the initial algorithm (3.1) has a second order accuracy in  $r$  while the last one (4.1) has a third order accuracy in  $r$ .

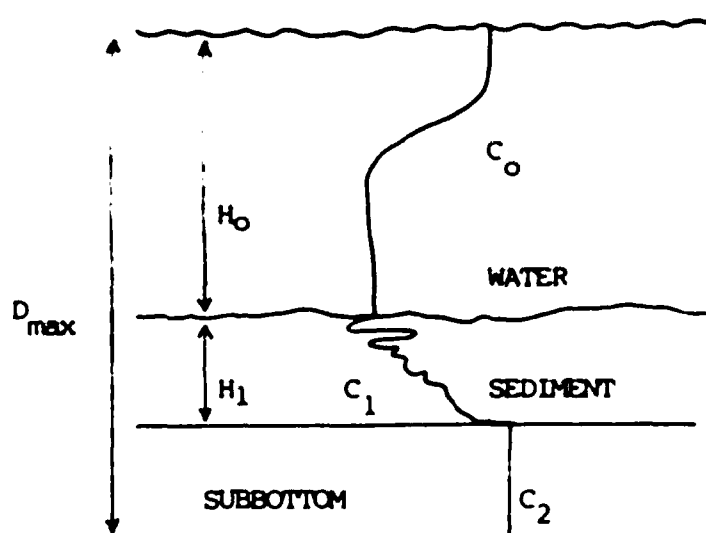
The Split Step Algorithm is a marching solution in range, therefore range dependence of the environmental data (sound velocity profiles, bathymetry) is easily introduced. Boundary conditions are implicit in the numerical solution. A pressure-release surface at the air/water interface is assumed, the field is made anti-symmetrical in depth about  $z = 0$  by using real sine transforms [10].

The Fourier theory assumes  $L^1$  integrable functions, the field must vanish below the maximum depth of the transform. A pressure release bottom must correspond to the maximum transform sample to avoid aliasing while doing FFTs.

An unphysical bottom layer is introduced to attenuate the field in the bottom [10]. This technique assures that at long range, the dominant contribution comes from low order modes that correspond to low grazing angle. The higher the mode, the greater is the grazing angle, and the more penetration into the bottom.

## 5. Description of MPL's PE Program

The algorithms used are the Split Step Algorithms (3.1) and (3.5) which have a second order accuracy in  $r$ . The initial transform size is determined by the input transform component  $N$ , the transform has a total of  $2^N$  points.  $N$  must be smaller than 12. The maximum depth of the transform  $Z_{max}$  is the maximum depth of the bottom  $D_{max}$  extended by a factor of 33 % to account for the unphysical attenuating layer.



If the input transform size is not given, it is determined by the program using

$$N = 1 + \log_2(kZ_{max}/2) \quad (5.1)$$

where  $k$  is the average wave number, and  $Z_{max}$  is the maximum depth transform.

The program begins by creating the starting field and, in a range loop, reads the environmental data, creates the refraction index table,  $e^{i\Delta r A}$ , and the second derivative table,  $e^{i\Delta r B}$ . If the range step is not fixed, the program computes a new range step at each range :  $\Delta r = \frac{\lambda}{1 - \cos \theta}$  where  $\lambda$  is the wavelength and

$$\theta = \tan^{-1} \left[ \frac{|sF(\Psi)|}{k_0 |F(\Psi)|} \right], \quad s \text{ is the Fourier variable, } F \text{ is the Fourier transform and } | |$$

is the  $L^2$  norm (i.e.  $\|F\| = \sqrt{\int F F^* ds}$ ).

If the new range step has a relative difference with respect to the old range step of at least 25% then this new range step is selected and a new table of second derivatives  $e^{i\alpha z}$  is computed. Otherwise, the old range step and the second derivative table are kept. The program does Fourier sine transforms and multiplies the transformed field by the table of second derivatives :

$$\exp\left[-i \frac{\Delta r_{\text{new}} + \Delta r_{\text{old}}}{2} \frac{s^2}{2k_0}\right] \quad (\text{standard PE}) \quad (5.2)$$

$$\exp\left[-i \frac{\Delta r_{\text{new}} + \Delta r_{\text{old}}}{2} \frac{\frac{s^2}{2k_0}}{\left(1 - \frac{s^2}{k_0^2}\right)^{1/2} + 1}\right] \quad (\text{modified PE}) \quad (5.3)$$

The inverse sine Fourier transform is performed and the field is then multiplied by

$$\exp\left[i \frac{\Delta r_{\text{new}}}{2} (n^2 - 1)\right] e^{-\alpha z} \quad (\text{standard PE}) \quad (5.4)$$

$$\exp\left[i \frac{\Delta r_{\text{new}}}{2} (n - 1)\right] e^{-\alpha z} \quad (\text{Modified PE}) \quad (5.5)$$

$\alpha$  is the attenuation in the water or in the bottom. The program advances the range to  $r = r_{\text{old}} + \Delta r_{\text{new}}$ .

After bookkeeping on the range dependence of the environment is completed, it checks for aliasing in the transform. Aliasing occurs when the contribution of steep angles in the angular spectrum is significant. The parabolic approximation is based on the "small angle" hypothesis. The angular spectrum is divided into four equal bins, their energy,  $E_1, E_2, E_3, E_4$  from low to high angles are computed. An aliasing factor is defined as  $10 \log_{10} R_1$  with

$R_1 = \frac{E_4}{E_1 + E_2 + E_3 + E_4}$ . If  $-14 \text{ dB} < R_1$ , then the energy is spread throughout a large

angular interval and aliasing is severe so the run is terminated. If  $-20 \text{ dB} < R_1 < -14 \text{ dB}$ , then a warning is issued and after five warnings the run is terminated. Two oversampling factors are defined,  $R_2 = \frac{E_3 + E_4}{E_1 + E_2}$  and  $R_3 = \frac{E_1}{E_2}$ . If  $R_2 < -70 \text{ dB}$  and  $R_3 < -60 \text{ dB}$ , the field is oversampled and the transform size is reduced by a factor of 2.

## 6. Phase Error Corrections

The parabolic approximation and the Split Step Algorithm are applicable if the index of refraction locally varies only slightly and if propagation is limited in a narrow aperture about the outward range.

Comparison between the parabolic equation and elliptic equation normal mode phase velocities in range independent waveguides [11,12] showed that the parabolic approximation introduces phase errors.

In the case of a single normal mode describing the propagation, the phase error is canceled if the reference wave number,  $k_0$ , is chosen to be the mode wave number [11]. When several modes are propagating, there is no obvious a priori choice for  $k_0$  [13] even if it can be intuitively understood as a weighted sum of the excited mode wave numbers [12]. In practice, the choice of  $k_0$  is left to the model user. Two different methods implemented on MPL's PE program try to correct these phase errors.

The first method by Brock et al [14], also called the CMOD correction, modifies the environment, i.e. the profile of the refraction index to minimize the phase errors. The idea is to decompose the sound field calculated by the parabolic equation into propagating normal modes and to compare these modes with those obtained by a WKB approximation to the elliptic wave equation. Requiring that the depths of the modal turning points (where the mode amplitudes are the largest) are the same in the parabolic and elliptic cases, that the phase velocities at the turning points are the same and assuming that there is no isovelocity region, a mapping can be derived :

$$(n, z) \rightarrow (\sqrt{n^2-1}, z\sqrt{n}) \quad (6.1)$$



The CMOD correction appears to offer improvements when the source and the receiver are near the surface, the bottom, or in regions near individual mode turning points.

Pierce introduced a better approach to determine in a more natural way the reference wave number by using the Rayleigh Principle : "if the propagation is progressive such that the energy is being transported on the average in one direction without propagation in the backward direction then the average kinetic energy is equal to the average potential energy" [13].

The kinetic energy in the volume element,  $V_o$ , is  $\frac{1}{2}\rho(|u|^2 + |v|^2)V_o$  where  $u, v$  are the particle velocities in the  $r, z$  directions respectively. The potential energy is  $-\int P dV$  where  $P$  is the pressure and  $dV$  is the change of volume of the volume element with varying pressure. If the volume element dimensions are small compared to the wavelength, the plane wave approximation can be used and  $V = V_o(1 - \frac{P}{\rho c^2})$  and therefore,  $dV = -\frac{V_o dP}{\rho c^2}$ . Then, the potential energy becomes  $\frac{P^2 V_o}{2\rho c^2}$  [15].

Using a volume element integrated over  $z$ , the Rayleigh Principle becomes

$$\int \frac{1}{2} \rho (|u|^2 + |v|^2) dz = \int \frac{1}{2} \frac{P^2}{\rho c^2} dz \quad (6.2)$$

This principle can be applied in underwater acoustics where the field is a superposition of modes [13]. The equation becomes

$$k_o^2 = \frac{\int (\frac{\omega}{c})^2 \rho^{-1} |\Phi|^2 dz - \int \rho^{-1} |\Phi_z|^2 dz}{\int \rho^{-1} |\Phi|^2 dz} \quad (6.3)$$

since

$$|u|^2 = \left[ \frac{k_o}{\omega \rho} \right]^2 \frac{|\Phi|^2}{r} \quad (6.4)$$

$$|v|^2 = \left( \frac{1}{\omega \rho} \right)^2 \frac{|\Phi_z|^2}{r} \quad (6.5)$$

$$|P|^2 = \frac{|\Phi|^2}{r} \quad (6.6)$$

$$\Phi_z = \frac{\partial \Phi}{\partial z} \quad (6.7)$$

This wave number is range dependent and can be proved to be range independent in range independent media. The method also allows discontinuities of density. The natural choice for  $k_z$  as given by Pierce consistently yielded more accurate results than when other schemes were used [13]. According to [1], it is strongly recommended, to ensure maximum accuracy of the PE calculations, to use this phase velocity correction together with the modified parabolic equation (Thomson and Chapman). MPL's PE program computes  $k_z$  every ten range steps.

## 7. Source Field

Initial fields must be computed to start the marching solution in range. The wave equation to solve is

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r} \frac{\partial p}{\partial r} + \frac{\partial^2 p}{\partial z^2} = -\frac{2p_o}{r} \delta(r) \delta(z-z_o) \quad (7.1)$$

where  $p_o$  is the pressure at one meter [6].  $p_o$  is set to unity.

The starting field can be either created by source routines available in the program or entered as an input in a file. The three available sources are the Gaussian source [6], the Greene source [16] and the Thomson source [7].

The Gaussian source is an asymptotic approximation to spherical spreading in the far field. The standard parabolic equation is simplified in the near field by neglecting the refraction term ;

$$2ik_o \frac{\partial \Psi}{\partial r} + \frac{\partial^2 \Psi}{\partial z^2} = 0 \quad (7.2)$$

with  $p = r^{-1/2} e^{ik_o r} \Psi$ . Near the point source at depth  $z_s$ , the field is spherically spreading  $p = \frac{e^{ik_o R}}{R}$  with  $R = \sqrt{r^2 + (z-z_s)^2}$ . To do an asymptotic matching between the two expression for  $p$ , one lets  $r \rightarrow \infty$  in the spherical spreading solution and, after a first order expansion :

$$p = \frac{1}{r} e^{ik_o r \left(1 + \frac{(z-z_s)^2}{2r^2}\right)} \left[1 - \frac{1}{2} \frac{(z-z_s)^2}{r^2}\right] \quad (7.3)$$

Taking the solution of the simplified parabolic equation, one lets  $r \rightarrow 0$ , and by asymptotic connection one gets

$$\Psi(0, z) = \frac{1}{\sqrt{r}} e^{\frac{ik_o (z-z_s)^2}{2r}} \quad (7.4)$$

This solution given by Tappert is simple and avoids spurious sidelobes [5,6].

This source has been modified to allow some tilt with respect to the horizontal.

The half width  $\theta_1$  of the source (half angle corresponding to a 3 dB loss with

respect with the center of the source) can be selected too. Taking  $\theta_1 = 0$  corresponds to the original source or the standard Gaussian source, that is to a half width of  $35^\circ$ .

The Greene source is documented in [16]. It has been designed for a higher angle parabolic equation (HAPE). It is a wide angle source with a  $-3 \text{ dB}$  half width of  $80^\circ$ .

The Thomson source is an alternate starting field that uses a Kaiser Bessel window to design a source that insonifies a limited aperture in the  $k_x$  space. The parameters describing this source are the depth of the source  $z_s$ , the half width of the source aperture  $\theta_1$ , the tilt of the source  $\theta_2$  and  $\alpha$  the level in  $\text{dB}$  between the stopband and the passband in the  $k_x$  space. The field is

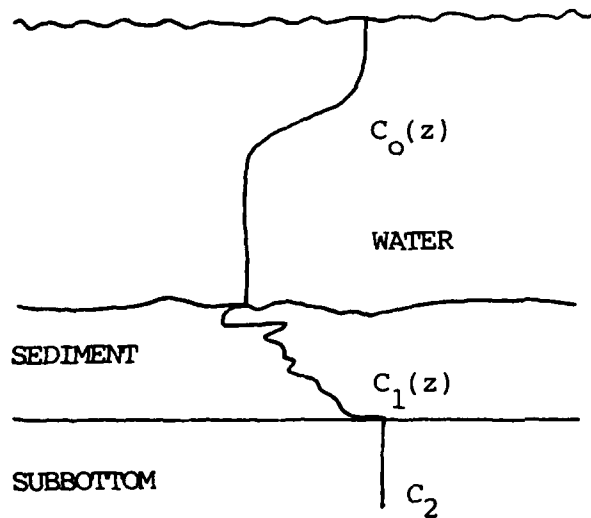
$$\Psi(0,z) = \left(\frac{c_o}{f}\right)^{1/2} \frac{1}{\pi(z-z_s)} \sin[2\pi \frac{f}{c_o} (z-z_s) \tan\theta_1] \frac{I_o(xy)}{I_o(x)} e^{2\pi \frac{f}{c_o} (z-z_s) \sin\theta_1} \quad (7.5)$$

with  $x = 0.1102 (\alpha - 21)$  and  $\alpha > 50$ . The Thomson source as implemented in PE uses  $\alpha = 60$  so  $x = 4.2978$ .  $I_o$  is the modified Bessel function of  $0^{\text{th}}$  order.  $y = (1 - [\frac{z-z_s}{Z_{\max}-z_s}]^2)^{1/2}$  where  $Z_{\max}$  is the maximum transform depth and  $c_o$  is the reference sound speed.

The PE program can also be started by a source field stored in a file. The file must have  $2^N$  pairs of the real and imaginary parts of the pressure. Each pair corresponds to the pressure field at a point of the mesh which has a depth increment equal to  $\frac{Z_{\max}}{2^N - 1}$ , where  $Z_{\max}$  is the maximum depth of the transform. This field should be normalized such that the field corresponds to a unit pressure source at one meter, and be a solution of (7.1). The Gaussian source and the Thomson source have been properly normalized.

## 8. Description and Treatment of the Environment

The treatment of the environment is described in the documentation of the SACLANT PE program [1]. The bottom is a two layer fluid bottom : a sediment layer characterized by a height, an arbitrary sound velocity profile, a density, and a compressional wave attenuation, and a subbottom layer characterized by a sound velocity, a density and a compressional wave attenuation.



The change of density between the various layers (water, sediments, subbottom) is taken care of by the method given by Tappert [6]. The correct wave equation is

$$\rho \nabla \left( \frac{1}{\rho} \nabla p \right) + \frac{\omega^2}{c^2} p = 0 \quad (8.1)$$

where  $\rho$  is the density. By doing the change of variable  $q = \frac{p}{\sqrt{\rho}}$ , the equation becomes

$$\nabla^2 q + k_o^2 n^2 q = 0 \quad (8.2)$$

where  $k_o = \frac{\omega}{c_o}$  and  $n$  is an effective index of refraction :

$$n^2 = \frac{c_o^2}{c^2} + \frac{1}{2k_o^2} \sqrt{\rho} \nabla \left( \rho^{\frac{-3}{2}} \nabla \rho \right) \quad (8.3)$$

The discontinuities of the density are not allowed and the program smears out its jumps. If the density jumps from  $\rho_1$  to  $\rho_2$ , the smoothed density is

$$\rho(z) = \frac{(\rho_1 + \rho_2)}{2} + \frac{(\rho_2 - \rho_1)}{2} \tanh\left(\frac{z-h}{L}\right) \quad (8.4)$$

and  $L$  is taken such that  $L = \frac{2}{k_o}$ .

The subbottom has a height twice the wavelength. If the phase velocity of the subbottom is set to zero, this height is also zero and there is no subbottom.

Bottom loss is introduced by adding a small imaginary part to the wave number [2],  $k = k_R + i\alpha$  with  $\alpha \ll k_R$  where  $k_R = \frac{\omega}{c}$ . This attenuation is expressed as part of the refraction index  $n^2 = (\frac{c_o}{c})^2 + i(\frac{c_o}{c})^2 \frac{2c\alpha}{\omega}$ , assuming that  $\alpha \ll \frac{\omega}{c}$ . The imaginary part of the refraction index is then expressed in term of the real part  $n_I^2 = \frac{2\alpha c}{\omega} n_R^2$ . The actual input for the attenuation is  $\beta$  in  $dB/\lambda$  where  $\lambda$  is the wavelength. Then  $\alpha = \beta \frac{\omega}{2\pi c (20 \log e)}$  and  $n_I^2 = \frac{\beta}{27.287527} n_R^2$ .

Volume attenuation in the water is introduced by multiplication of the field by a term  $e^{-\alpha r}$  [1] but this term is fairly low at the frequencies of interest for PE, that is below 200 Hz.

Since the environment can vary in range, one must provide the program with sets of the environmental data at specified ranges. A sector is defined by these ranges associated with environmental data. In each sector, the sound velocity profile is range independent while the bathymetry is linearly interpolated in range between the two ends of the sector (water depth, sediment depth, subbottom depth). The sound velocity profile is interpolated linearly with depth. However the sound velocity profile will jump from one sector to another because it is not interpolated in range.

## 9. Program Inputs

The program has a readable input file. Data can be divided into two groups. One has the general information such that frequency, source depth, filenames for output results. The other contains environmental information that must be repeated for each sector. A list of the inputs follows. All of the following information must be included or lines left blank.

A1. Title of the input file or run

A2. five output filenames of seven characters each where the output will be dumped (sector information file, transmission loss versus range at selected receiver depths, transmission loss versus depth at selected ranges, SIO data file for subsequent use for contour plot and SIO outfilename for the pressure field). If blank, no file is generated.

A3. Transform size (integer smaller or equal to 12), 0 meaning that the program itself will compute the transform size based on a sufficient sampling for the Gaussian source.

A4. Reference sound speed (allowing the calculation of  $k_o$ ). A negative sound speed means that the Pierce phase correction will be used. A nul sound velocity means that the CMOD correction will be used. Otherwise the given sound velocity will be used as reference sound velocity in the calculation of the index of refraction, and its value must have a physical value (e.g. 1500  $m/s$ ).

A5. Output range step size (in  $m$  ). It must be greater than the range step.

A6. Frequency (in  $Hz$  ) since at high frequency the range step is small, the frequency should be taken below 200  $Hz$  .

A7. Source depth (in  $m$ )

A8. Maximum range of track (in km)

A9. Parabolic equation flag ; 0 means that the standard parabolic equation will be used, 1 means that the modified parabolic equation will be used.

A10. Spreading flag ; 0 means that the source is a point source in a 3-D geometry while 1 means that the source is a line source in a 2-D geometry (i.e. a horizontal line source perpendicular to the range-depth plane).

A11. Source flag ; 0 means that the starter will be the Gaussian source, 1 the starting field is provided by the user in a file, 2 the starter is the Greene source and 3 the starter is the Thomson source.

A12. Half beam width and Beam tilt (up is positive) with respect to the horizontal (in degrees) for source flag equal to 0, 2, 3.

A13. Filename containing the starting field when it is provided as input (source flag equal 1).

A14. Number of receiver depths and number of selected ranges (each one  $\leq 20$ ) where transmission loss is desired.

A15. Number of ranges and number of depths for the contour plot (maximum dimension of 100 for each).

A16. List of the receiver depth array (m) in increasing order

A17. List of the selected ranges (m) in increasing order

A18. Number of sectors (the environmental data must be repeated for each sector)

B1. Water depth (in m) and starting range of the sector (in km)

B2. Number of points in the water sound velocity profile



- B3. List of pairs depth, sound velocity (in m and m/s)
- B4. Sediment height in m and density (in  $g/cm^3$ )
- B5. Attenuation in the sediments ( in  $dB/\lambda$  )
- B6. Number of points in the sediment sound velocity profile
- B7. List of the couples height, sound velocity (in m and m/s)
- B8. Subbottom density (in  $g/cm^3$ ), attenuation (in  $dB/\lambda$ )
- B9. Subbottom sound velocity (in m/s)

When two arguments are on the same line, be sure to have at least one space (not a comma) in between. The couples height/sound velocity must be such that the first sound velocity value corresponds to a zero height and the last one to the maximum height. Also note that heights are used and not depths for the sediment layer.

## 10. Program Output

The user is allowed in the input file to ask for five different outputs :

- a sector information file that summarizes all environmental information
- a transmission loss versus range file that contains the transmission loss versus range for each receiver depth over the entire range of track. A multi-channel SIO data file is automatically created for subsequent processing and plotting. The name of this SIO data file is the inputted filename with the "sio" extension. The first channel of the file contains the ranges and the following ones (up to 20) correspond to a receiver depth and contain the transmission loss.
- a transmission loss versus depth that contains the transmission loss versus depth for each selected range, over the maximum physical depth. A multi-channel SIO data file is automatically created for subsequent processing and plotting. The name of this SIO file is the inputted filename with a "sio" extension. The file is structured in pair of channels where the  $2p-1^{th}$  and the  $2p^{th}$  channel correspond to the depth and the transmission loss of the  $p^{th}$  selected range.
- a SIO data file to feed a contour routine. This file has a number of channels equal to the number of depth samples (A15). Each channel has a number of points equal to the number of ranges samples (A15). The file is an equi-spaced grid in range and depth of transmission loss values.
- a SIO data file containing the pressure field where the first channel contains the real part of the pressure and the second one the imaginary part of the pressure. This output requires a fixed range step and a fixed transform size  $N$ . Each channel is made of sequential  $2^N$  point segments ; Each segment corresponds to a range sample.

## 11. Examples

### 11.1. Isovelocity Shallow Water (Range Independent)

The first example is a shallow water run taken from [1]. The medium is range independent with isovelocity water and bottom, but without subbottom. The source used is the standard Gaussian source (half bandwidth of  $35^\circ$ ) at a depth of 25 m. The frequency is 50 Hz. The standard parabolic equation is used with a reference sound speed equal to 1500 m/s. No phase velocity correction is applied. A copy of the script, the input file and the plot control files are presented below. In this example two modes are propagating, one with a stronger attenuation than the other. We find the same transmission loss versus range plot as in [1]. This run takes 15 minutes on a Sun Workstation 3/50 (Sun Workstation is a registered trademark of Sun Microsystems, Inc).

Apr 19 16:40 1988 scr Page 1

```
#
# EXAMPLE 1
#
pe << INN
example.1
INN
#
# transform the range from meter unit into km unit
#
demux 1 outfil2sio temp1
demux 2 outfil2sio temp2
multc 0.001 temp1 temp
mux temp temp2 outfil2siobis
#
rm temp temp1 temp2
#
2n << INN > example1.plt1
t1ran.con
n
outfil2siobis
outfil2siobis
INN
#
plotwin example1.plt1
#
```

```

# input file for the program pe (MPL's pareq version)
# title of the input data file
Sacilant Test #1 : Shallow Water
# enter the 6 file names to save the output data
##### 7 characters is the required format
outfill # sectors information file
outfill2 # transmission loss versus range at selected depths
# transmission loss versus depth at selected ranges
# contour data in sio data file
# filed data sio data file
# transform size (0 means that pe takes care of it)
8
# range step (0 means it is variable in pe)
10
# reference sound speed in m/s (can be 0, <0, >0)
1500
# output range distance in m
50
# frequency in Hz
50
# source depth in m
25
# maximum range of track in km
20.
# operational mode of parabolic equation model (0 or 1)
1
# flag for geometry (0=3-D point source,1=2-D line source)
0
# source 0=gaussian,1=field given in file,2=green,3=thompson
0
# half beam width, beam tilt from horizontal in degrees
0.
0.
# input filename containing the starting field (source=1)
# number of receiver depths, of selected ranges for tl vs range or depth
1
0
# number of ranges, number of depths for contour plot
0
0
# input receiver depth array for tl vs range in m
25.
# input selected range array for tl vs depth in m
20000.
# number of sectors ( integer)
1
1
# repeat for each sector :
# water depth in m, starting range in km
100.
0
# number of points in the water sound velocity profile ( integer)
2
# sound velocity profile : depth in m, velocity in m/s
0.
1500.
100.
1500.
# sediment height in m, sediment density in g/cm**3

```

```

250.
1.5
# attenuation in sediment in dB/wavelength
0.5
# number of points in the sediment sound velocity profile ( integer)
2
# sound speed profile in sediment : depth in m, velocity in m/s
0.
1600.
250.
1600.
# subbottom density in g/cm**3 and attenuation in dB/wavelength
0.
0
# sound velocity in subbottom m/sec
0.

```

Feb 5 13:05 1988 outfill Page 1

SOUND SPEED PROFILE IN WATER # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1500.00
100.00	1500.00

SOUND SPEED PROFILE IN SEDIMENT # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1600.00
250.00	1600.00

SEDIMENT DENSITY = 1.500  
SEDIMENT ATTENUATION = 0.500  
NO SUBBOTTOM

GLOBAL PLOT CONTROL PARAMETERS

-1, numplt- number of data files plotted on one plot (2)

8.0, xlen- x axis length in inches (3)

1, noaxe - axis suppression. (0) suppress, (1) box, (2) lower, (3) mid (4)

0, nsf- sampling frequency flag (5)

0, nexpx- x axis exponential control (6)

# INDEPENDENT X AXIS PARAMETERS

0.0, xorg- x axis origin in user units (8)

1, xinc- x axis sample increment in user units (9)

# AXIS SCALING PARAMETERS

20., xmax- x maximum in user units (11)

0, xmin- x minimum in user units (12)

5., xtic- x tic mark spacing in user units (13)

3, numbers parallel (14)

.125 size (15)

2, justify (16)

1, nxp- number of places after the decimal point (17)

# FILE PARAMETERS

# fillin- input d.a. file name (19)

SETUP FOR INDIVIDUAL PLOT

1.5, x3 - lower left corner x position (2)

1.0, y3 - lower left corner y position (3)

5.0, ylen - length of axis (4)

40.0, ymax (5)

100.0, ymin (6)

10., ytic (7)

3, nrot (8)

.125, size (9)

2, dec justify (10)

1, nyp - number of places after decimal place (11)

1, nlog - data manipulation control (12)

0, nexpy- y axis exponential control (13)

# YSTRNG - on axis (14)

Loss (dB) (16)

# desired MAJOR TITLE (16)

Example 1 : Shallow Water (8,10,50,1500) T.C. (18)

# desired MINOR TITLE (18)

Freq.: 50 Hz, SD: 25 m, RD: 25 m, GAUSS (0,0) (20)

0.20, titsiz - title size (21)

1.5, xloc - lower left x location (22)

6.7, yloc - lower left y location (23)

# desired Y TITLE (90 degrees to horizontal) (23)

0.8, xtloc - lower right x position (25)

2.0, ytloc - lower y position (26)

0.25, ytsiz- y title size (27)

# AXIS LABEL PARAMETERS

# XSTRNG- x axis label string (29)

Range (km)

2, chan1 - channel number to plot (31)

1, chan2 - channel to do math with (32)

-1, nreal- flag for plotting real or imaginary part-also da (33)

1, ndp0- first point plotted for partial file plotting (34)

0, ndtot- total number of points plotted for partial plot. (35)

1, nskip- plotting point increment (36)

0, numsyn- symbol number for point plotting (37)

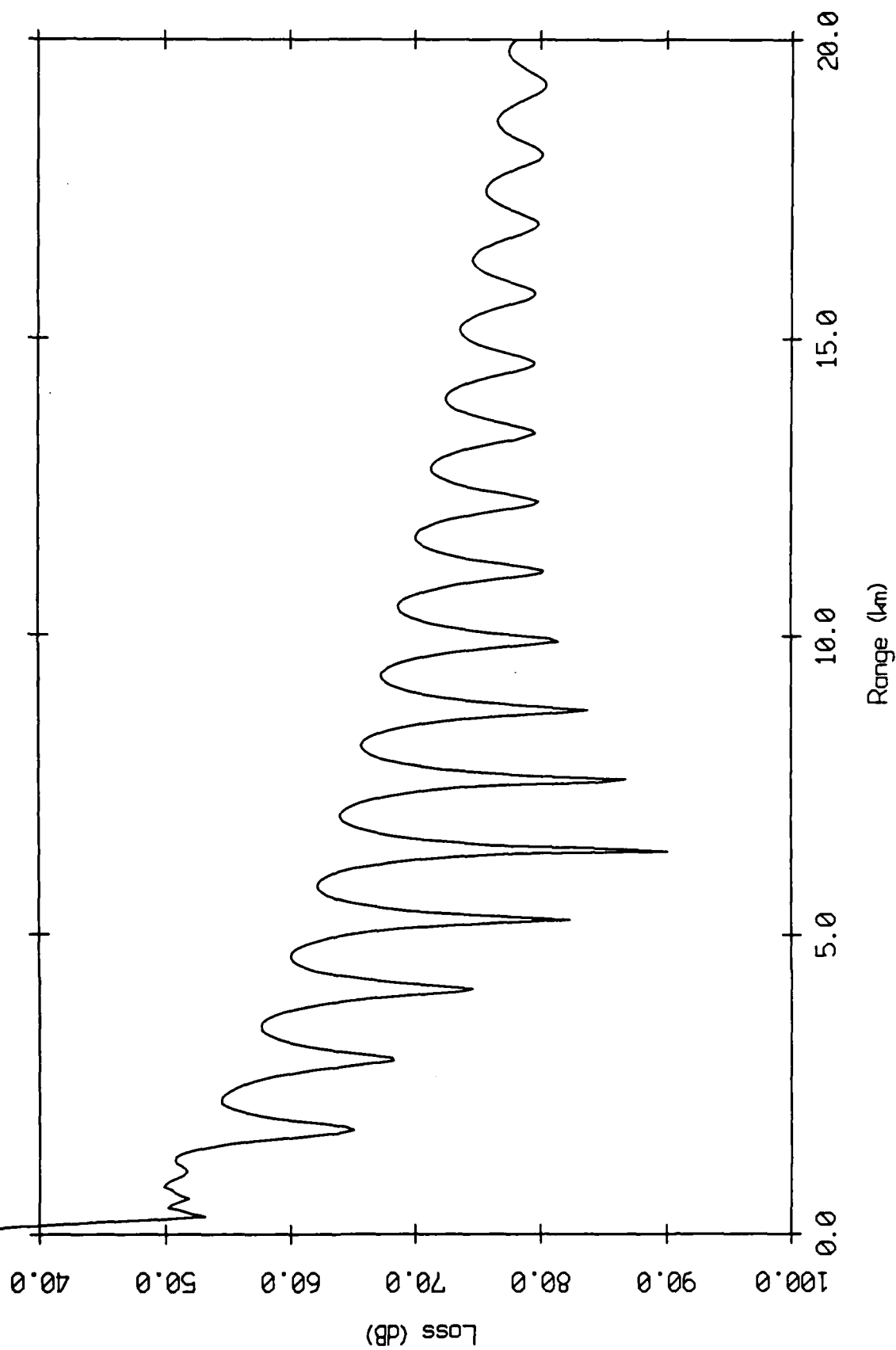
0.15, symsiz- symbol size in inches (38)

0.0, xoff- x offset in user units (39)

0.0, yoff- y offset in user units (40)

0, ndash - number of dashed lines for this plot (41)

Example 1 : Shallow Water (8, 10, 50, 1500) T.C.  
Freq.: 50 Hz, SD: 25 m, RD: 25 m, GAUSS (0, 0)





### 11.2. Wedge-Shaped Ocean with Penetrable Bottom

The second example tries to repeat the results by Jensen and Kuperman [17]. The environment consists of an initial 5 km stretch of 200 m depth followed by a bottom slope of  $1.55^\circ$ . The source used is the standard Gaussian source (the half width is not given in [17]), at 112 m depth, the frequency is 25 Hz. The maximum water depth is 200 m with a sediment layer extending to a depth of 800 m.

The program is such that the contour plot extends in range from the source to the maximum range, and in depth from the surface to the maximum depth. The maximum depth is the maximum sum of the water depth, sediment/bottom and subbottom heights.

The transmission loss versus depth extends in depth from the surface to the maximum depth like in the contour plot, and all mesh points are used. The transmission loss versus depth is interpolated in range when the selected range is not a range mesh point.

The contour plot is presented below. One finds good agreement with [17] although the contour routines are different. A copy of the input file as well as the script and plot control files and the sector information output file are following. This job takes 50 minutes to run on a Sun Workstation 3/50 (Sun Workstation is a registered trademark of Sun Microsystems, Inc).

Apr 20 08:59 1988 script Page 1

```
#  
#  
# EXAMPLE 2  
#  
# /localdata/jean/pe/pe << INN  
example.2  
INN  
#  
# plot the CONTOUR  
#  
# contour <(INN > contour_plot1  
plotconl  
outfil4  
n  
# INN  
#  
# plotwin contour_plot1  
#  
#
```

```

# input file for the program pe (MPL's pareq version)
# title of the input data file
Sacant Test coastal slab
##### 6 file names to save the output data
##### 5 characters is the required format
outfil0 # sectors information file
# transmission loss versus range at selected depths
outfil4 # sio data file for contour
# sio data file for the field
# transform size (0 means that pe takes care of it)
10
# range step (0 means it is variable in pe)
10
# reference sound speed in m/s (can be 0, <0, >0)
1500
# output range distance in m
50
# frequency in Hz
25
# source depth in m
112.
# maximum range of track in km
15.
# operational mode of parabolic equation model (0 or 1)
0
# flag for spreading (0-cylindrical 1-spherical)
0
# source 0-gaussian, 1-field given in file, 2-green, 3-thompson
0
# half beam width, beam tilt from horizontal in degrees
0.
# input filename containing the starting field (source=1)
# number of receiver depths, of selected ranges for tl vs range or depth
1
# number of ranges, number of depths for contour plot
100 100
# input receiver depth array for tl vs range in m
112.
# input selected range array for tl vs depth in m
8000.
# number of sectors ( integer)
3
# repeat for each sector :
#
# water depth in m, starting range in km
200. 0
# number of points in the water sound velocity profile ( integer)
2
# sound velocity profile : depth in m, velocity in m/s
0. 1500.
260. 1500.
# sediment height in m, sediment density in g/cm**3

```

```

600. 1.15
# attenuation in sediment in dB/wavelength
0.5
# number of points in the sediment sound velocity profile ( integer)
2
# sound speed profile in sediment : depth in m, velocity in m/s
0. 1704.5
600. 1704.5
# subbottom density in g/cm**3 and attenuation in dB/wavelength
1.15 .5
# sound velocity in subbottom m/sec
1704.40.
# water depth in m, starting range in km
200. 5.0
# number of points in the water sound velocity profile ( integer)
2
# sound velocity profile : depth in m, velocity in m/s
0. 1500.
200. 1500.
# sediment height in m, sediment density in g/cm**3
600. 1.15
# attenuation in sediment in dB/wavelength
0.5
# number of points in the sediment sound velocity profile ( integer)
2
# sound speed profile in sediment : depth in m, velocity in m/s
0. 1704.5
600. 1704.5
# subbottom density in g/cm**3 and attenuation in dB/wavelength
1.15 .5
# sound velocity in subbottom m/sec
1704.40.
# water depth in m, starting range in km
1. 12.50
# number of points in the water sound velocity profile ( integer)
2
# sound velocity profile : depth in m, velocity in m/s
0. 1500.
1. 1500.
# sediment height in m, sediment density in g/cm**3
799. 1.15
# attenuation in sediment in dB/wavelength
0.5
# number of points in the sediment sound velocity profile ( integer)
2
# sound speed profile in sediment : depth in m, velocity in m/s
0. 1704.5
799. 1704.5
# subbottom density in g/cm**3 and attenuation in dB/wavelength
1.15 .5
# sound velocity in subbottom m/sec
1704.40.

```

## SOUND SPEED PROFILE IN WATER # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1500.00
200.00	1500.00

## SOUND SPEED PROFILE IN SEDIMENT # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1704.50
600.00	1704.50

SEDIMENT DENSITY = 1.150  
SEDIMENT ATTENUATION = 0.500  
NO SUBBOTTOM

## SOUND SPEED PROFILE IN WATER # 2 AT RANGE 5000.0 M

DEPTH(M)	SPEED(M/S)
0.00	1500.00
200.00	1500.00

## SOUND SPEED PROFILE IN SEDIMENT # 2 AT RANGE 5000.0 M

DEPTH(M)	SPEED(M/S)
0.00	1704.50
600.00	1704.50

SEDIMENT DENSITY = 1.150  
SEDIMENT ATTENUATION = 0.500  
NO SUBBOTTOM

## SOUND SPEED PROFILE IN WATER # 3 AT RANGE 12500.0 M

DEPTH(M)	SPEED(M/S)
0.00	1500.00
1.00	1500.00

## SOUND SPEED PROFILE IN SEDIMENT # 3 AT RANGE 12500.0 M

DEPTH(M)	SPEED(M/S)
0.00	1704.50
799.00	1704.50

SEDIMENT DENSITY = 1.150  
SEDIMENT ATTENUATION = 0.500

NO SUBBOTTOM

```

# Contour draft plot control file
# Labels
# title - label centered above plot
# caption - label at bottom left corner of plot ( left justified )
#
# RANGE in KM      xlab - x axis label - centered at bottom of plot
# DEPTH in M       ylab - y axis label - centered at left side of plot
# Label Sizes
# .15,      titsiz - Size of Title
# .15,      capsiz - Size of Caption
# .15,      xlabsz - Size of x axis label
# .15,      ylabsz - Size of y axis label
# .10,      levsiz - Size of level labels
# 1,        scalef - Scale of level labels (level value / scalef)
# Data Manipulation
# 100,      colnum - # of columns for the input array (neg = flip vertical)
# -100,     rownum - # of rows for the input array (neg = flip horizontal)
# 0,        rotnum - number of 90 degree clockwise rotations
# Plot Axis Parameter
# 8.0,      length of the x-axis
# 5.0,      height of the y-axis
# 1.5,      x-org
# 1.0,      y-org
# Plot Parameters
# 0,         xmin - min x axis label value
# 15,        xmax - max x axis label value
# 800.,      ymin - min y axis label value
# 0.,        ymax - max y axis label value
# 16,        connum - number of contour levels (if 0 program will choose)
# 0,         codesw - toggle for line weighting. must have 'connum' of codes.
#           Contour levels if specified by user

```

Codesw weighting values

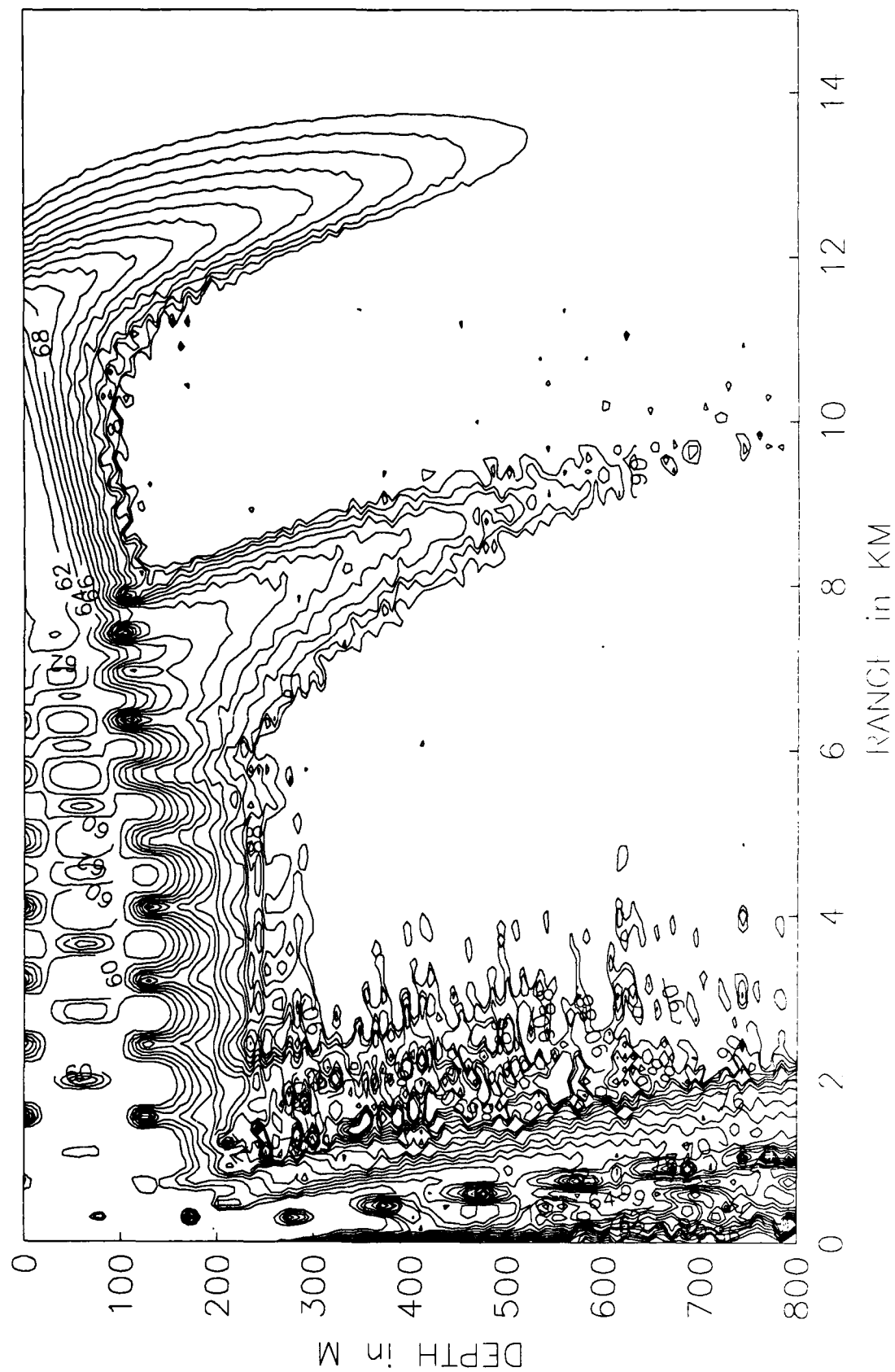
```

# 1 1
# 2 2
# 0 0
# 1 1
# 2 2

```

0

EXAMPLE 2 : COASTAL SLAB [17]



### 11.3. Deep Water Example I (Range Independent)

This example is of deep water propagation over a range of 60 *km*, the water depth is 2000 *m*, the frequency is 100 *Hz* and the source is a 2° narrow and 8° tilted Gaussian source at 250 *m* depth [1]. One finds once again agreement between the MPL's PE program and [1]. Script, input, plot control files and plots follow. This run takes 30 minutes on a Sun Workstation 3/50 (Sun Workstation is a registered trademark of Sun Microsystems, Inc).

Apr 20 08:39 1988 scr Page 1

```

#
# EXAMPLE 3
#
pe << INN
example.3
INN
#
# first demultiplex the channel 1 to yield the proper
# units in km for the plotting package and clip the
# transmission loss values between 70 and 130 dB
#
demux 1 outfilbsio temp1
demux 2 outfilbsio temp2
demux 3 outfilbsio temp3
multc .001 temp1 temp
clip 130 70 temp2 temp22
clip 130 70 temp3 temp33
mux temp temp22 temp1
mux temp1 temp33 outfilbsiobis
#
rm temp*
#
# plot
#
2n << INN > example3.plt1
tiran1.con
n
outfilbsiobis
outfilbsiobis
INN
#
plotwin example3.plt1
#
2n << INN > example3.plt2
tiran2.con
n
outfilbsiobis
outfilbsiobis
INN
#
plotwin example3.plt2
#
contour << INN > contourplot
pltcon1
outfld
n
INN
#
plotwin contourplot

```



```

# input file for the program pe (MPL's pareq version)
#
# title of the input data file
# Saciant Test #1 : Deep Water
# enter the 6 file names to save the output data
##### 7 characters is the required format
outfila # sectors information file
outfilb # transmission loss versus depth at selected depths
outfilc # transmission loss versus depth at selected ranges
outfild # sio data file for contour
outfile # sio field file
# transform size (0 means that pe takes care of it)
0
# range step (0 means it is variable in pe)
0
# reference sound speed in m/s (can be 0, <0, >0)
0
# output range distance in m
100
# frequency in Hz
100
# source depth in m
250
# maximum range of track in km
60
# operational mode of parabolic equation model (0 or 1)
0
# flag for spreading (0-cylindrical 1-spherical)
0
# source function (0-gaussian,1-mdal,2-greene,3-thompson)
0
# half beam width, beam tilt from horizontal in degrees
1.8
# lower and upper modes (integers)
0 0
# number of receiver depths, of selected ranges for tl vs range or depth
2 0
# number of ranges, number of depths for contour plot
100 100
# input receiver depth array for tl vs range in m
50.
# input selected range array for tl vs depth in m
250.
# number of sectors ( integer)
1
# repeat for each sector :
#
# water depth in m, starting range in km
2000 0
# number of points in the water sound velocity profile ( integer)
5
# sound velocity profile : depth in m, velocity in m/s
0. 1480.
100. 1476.

```

```

500 1470
1000. 1476.
2000. 1490.
# sediment height in m, sediment density in g/cm**3
100. 1.5
# attenuation in sediment in dB/wavelength
0.5
# number of points in the sediment sound velocity profile ( integer)
2
# sound speed profile in sediment : depth in m, velocity in m/s
0. 1475.
100. 1600.
# subbottom density in g/cm**3 and attenuation in dB/wavelength
2.0 0.1
# sound velocity in subbottom m/sec
1800.

```

Feb 5 14:06 1988 outfila Page 1

SOUND SPEED PROFILE IN WATER # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1480.00
100.00	1476.00
500.00	1470.00
1000.00	1476.00
2000.00	1490.00

SOUND SPEED PROFILE IN SEDIMENT # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1475.00
100.00	1600.00

SEDIMENT DENSITY - 1.500  
SEDIMENT ATTENUATION - 0.500

SUBBOTTOM DENSITY - 2.000  
SUBBOTTOM ATTENUATION - 0.100  
SUBBOTTOM SOUND SPEED - 1800.000

GLOBAL PLOT CONTROL PARAMETERS

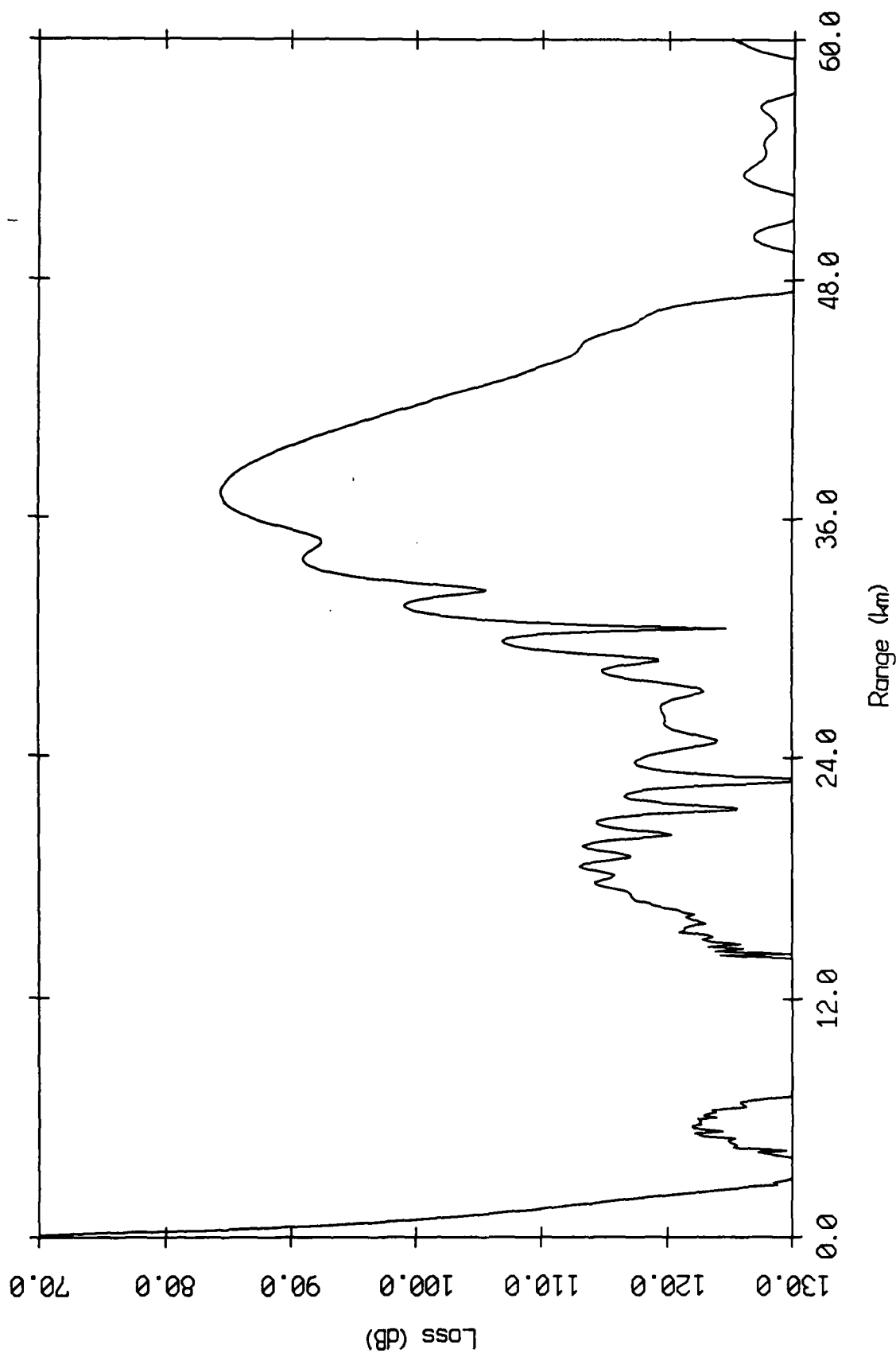
numplt- number of data files plotted on one plot (2)  
 xlen- x axis length in inches (3)  
 noaxe - axis suppression. (0) suppress, (1) box, (2) lower, (3) mid (4)  
 nsf- sampling frequency flag (5)  
 nexpx- x axis exponential control (6)  
 # INDEPENDENT X AXIS PARAMETERS (8)  
 xorg- x axis origin in user units (9)  
 xinc- x axis sample increment in user units (11)  
 # AXIS SCALING PARAMETERS (12)  
 60., xmax- x maximum in user units (13)  
 0, xmin- x minimum in user units (14)  
 12., xtic- x tic mark spacing in user units (15)  
 3, numbers parallel (16)  
 .125 size (17)  
 2, justify (19)  
 1, nxp- number of places after the decimal point (19)  
 # FILE PARAMETERS  
 # filin- input d.a. file name (19)

SETUP FOR INDIVIDUAL PLOT

1.5, x3 - lower left corner x position (2)  
 1.0, y3 - lower left corner y position (3)  
 5.0, ylen - length of axis (4)  
 70.0, ymax (5)  
 130.0, ymin (6)  
 10., ytic (7)  
 3, nrot (8)  
 .125, size (9)  
 2, dec justify (10)  
 1, nyp - number of places after decimal place (11)  
 1, nlog - data manipulation control (12)  
 0, nexpy- y axis exponential control (13)  
 # YSTRING - on axis (14)  
 Loss (dB) (16)  
 # desired MAJOR TITLE (18)  
 Example 3 : Deep Water (0,0,100,0) \*\* TILTED BEAM  
 # desired MINOR TITLE  
 Freq.: 100 Hz, SD: 250 m, RD: 50 m, GAUSS (2,8)  
 0.20, titsiz - title size (20)  
 1.5, xloc - lower left x location (21)  
 6.7, yloc - lower left y location (22)  
 # desired Y TITLE (90 degrees to horizontal) (23)  
 0.8, xtloc - lower right x position (25)  
 2.0, ytloc - lower y position (26)  
 0.25, ytsiz- y title size (27)  
 # AXIS LABEL PARAMETERS  
 # XSTRING- x axis label string (29)  
 Range (km)  
 2, chan1 - channel number to plot (31)  
 1, chan2 - channel to do math with (32)  
 -1, nreal- flag for plotting real or imaginary part-also da (33)  
 1, ndp0- first point plotted for partial file plotting (34)  
 0, ndtot- total number of points plotted for partial plot. (35)  
 1, nskip- plotting point increment (36)

0, numsyn- symbol number for point plotting (37)  
 0.15, symsiz- symbol size in inches (38)  
 0.0, xoff- x offset in user units (39)  
 0.0, yoff- y offset in user units (40)  
 0, ndash - number of dashed lines for this plot (41)

Example 3: Deep Water (0, 0, 100, 0) \*\* TILTED BEAM  
Freq.: 100 Hz, SD: 250 m, RD: 50 m, GAUSS (2, 8)



```

GLOBAL PLOT CONTROL PARAMETERS
numplt- number of data files plotted on one plot      (2)
xlen- x axis length in inches                        (3)
noaxe - axis suppression. (0) suppress, (1) box, (2) lower, (3) mid (4)
nsf- sampling frequency flag                          (5)
nexp- x axis exponential control                     (6)
# INDEPENDENT X AXIS PARAMETERS
xorg- x axis origin in user units                    (8)
xinc- x axis sample increment in user units          (9)
# AXIS SCALING PARAMETERS
xmax- x maximum in user units                       (11)
xmin- x minimum in user units                       (12)
xtic- x tic mark spacing in user units              (13)
numbers parallel                                     (14)
size                                                  (15)
justify                                              (16)
2, nxp- number of places after the decimal point   (17)
# FILE PARAMETERS
# filln- input d.a. file name                       (19)

```

```

SETUP FOR INDIVIDUAL PLOT
1.5, x3 - lower left corner x position              (2)
1.0, y3 - lower left corner y position              (3)
5.0, ylen - length of axis                          (4)
70.0, ymax                                         (5)
130.0, ymin                                         (6)
10., ytic                                          (7)
3, nrot                                             (8)
.125, size                                         (9)
2, dec justify                                    (10)
1, nyp - number of places after decimal place      (11)
1, nlog - data manipulation control                 (12)
0, nexpy- y axis exponential control               (13)
# YSTRING - on axis                               (14)
Loss (dB)                                          (16)
# desired MAJOR TITLE                             (18)
Example 3 : Deep Water (0,0,100,0) ** TILTED BEAM
# desired MINOR TITLE                             (20)
Freq.: 100 Hz, SD: 250 m, RD: 250 m, GAUSS (2,8)
0.20, titsiz - title size                          (21)
1.5, xloc - lower left x location                  (22)
6.7, yloc - lower left y location                  (23)
# desired Y TITLE (90 degrees to horizontal)
0.8, xtloc - lower right x position                 (25)
2.0, ytloc - lower y position                       (26)
0.25, ytsiz- y title size                           (27)
# AXIS LABEL PARAMETERS
# XSTRING- x axis label string                     (29)
Range (km)
3, chan1 - channel number to plot                   (31)
1, chan2 - channel to do math with                  (32)
-1, nreal- flag for plotting real or imaginary part-also da (33)
1, ndp0- first point plotted for partial file plotting (34)
0, ndtot- total number of points plotted for partial plot. (35)
1, nskip- plotting point increment                   (36)

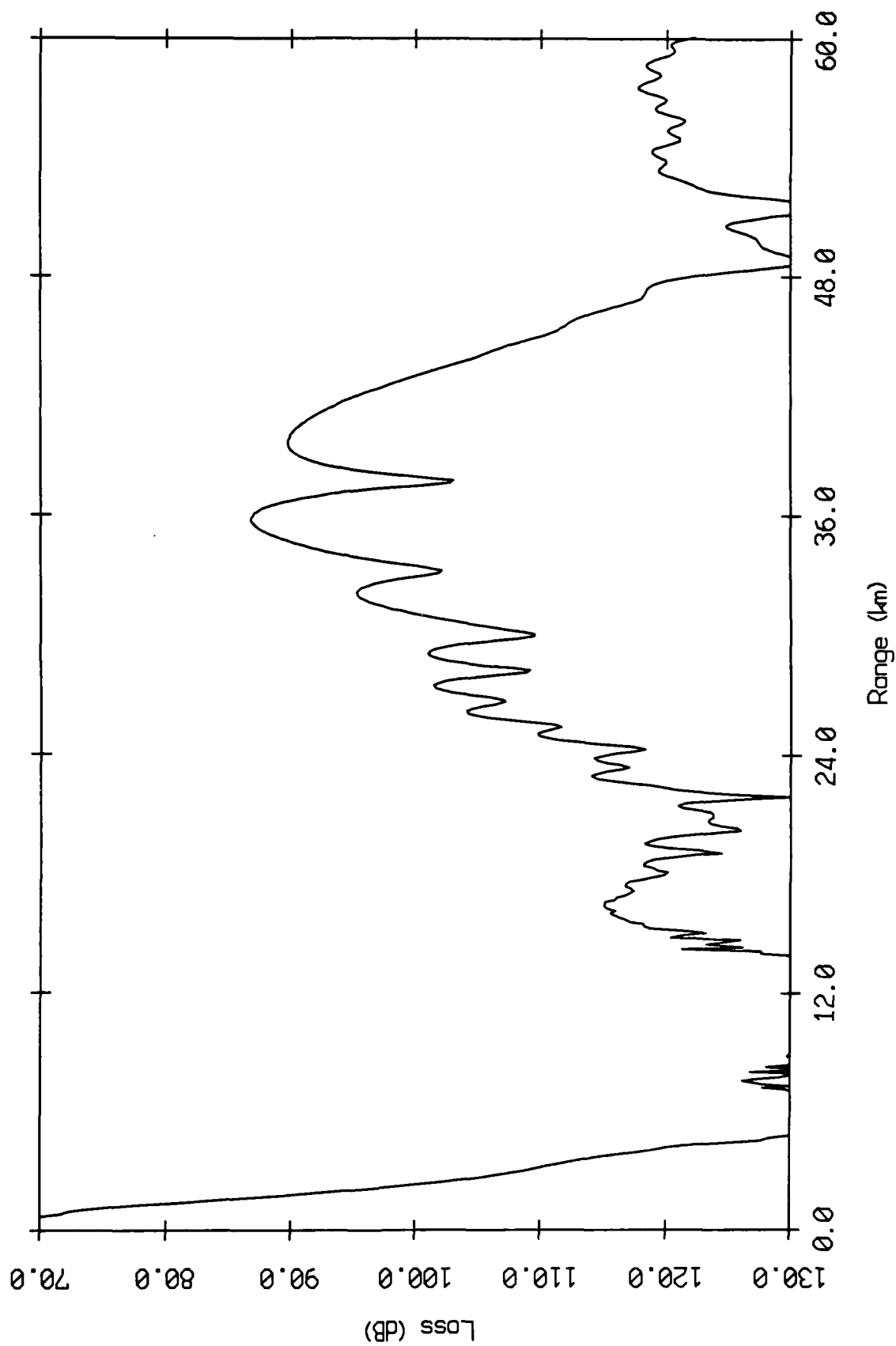
```

```

0, numsyn- symbol number for point plotting        (37)
0.15, symsiz- symbol size in inches                 (38)
0.0, xoff- x offset in user units                   (39)
0.0, yoff- y offset in user units                   (40)
0, ndash - number of dashed lines for this plot    (41)

```

Example 3 : Deep Water (0, 0, 100, 0) \*\* TILTED BEAM  
Freq.: 100 Hz, SD: 250 m, RD: 250 m, GAUSS (2, 8)



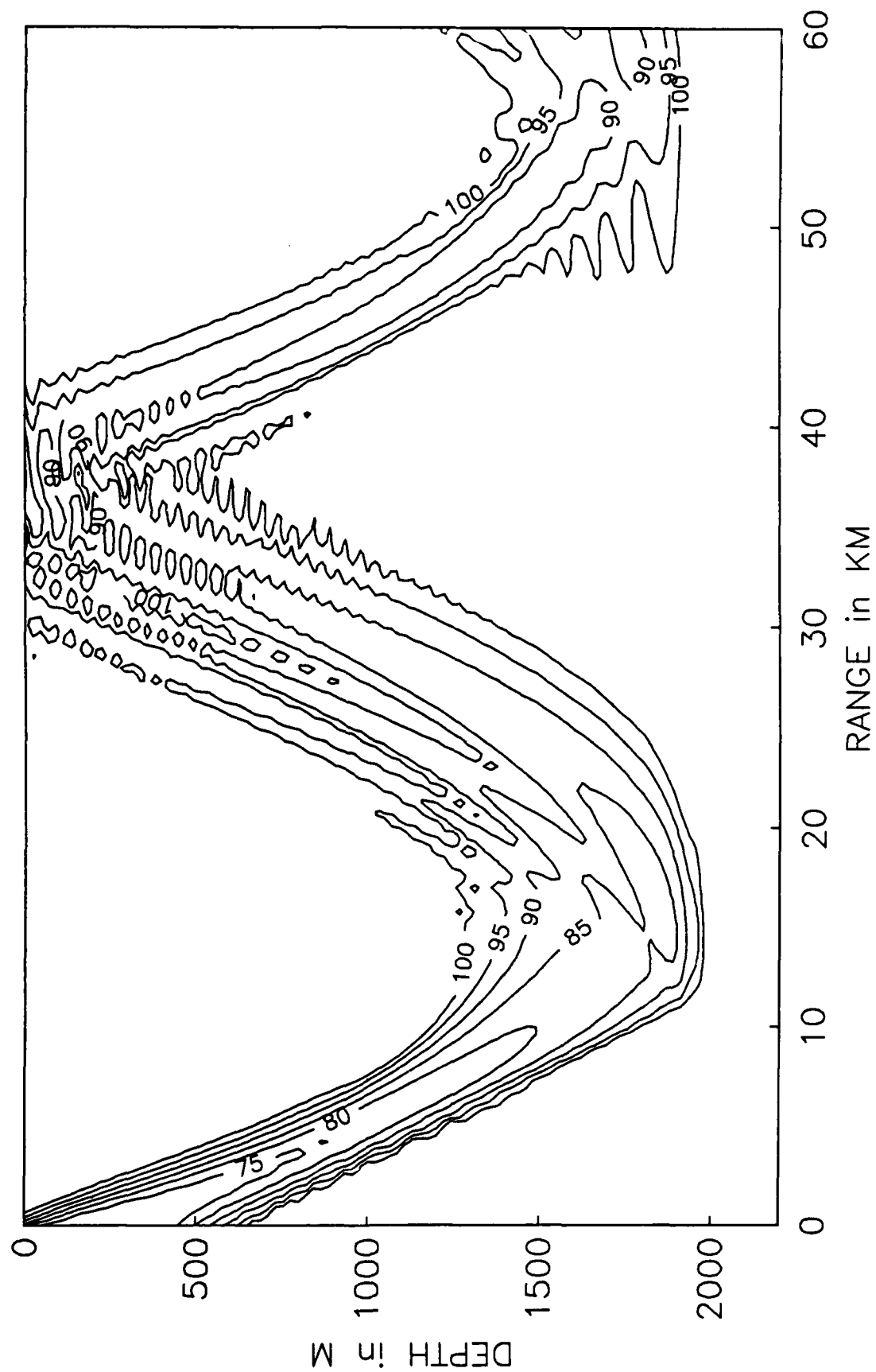
Apr 19 17:37 1988 pltconl Page 1

```

# Contour draft plot control file
# Labels
# title - label centered above plot
# caption - label at bottom left corner of plot ( left justified )
#
# RANGE in KM
# DEPTH in M
# Label Sizes
# .15, titsiz - Size of Title
# .15, capsiz - Size of Caption
# .15, xlab - Size of x axis label
# .15, ylab - Size of y axis label
# .10, levsiz - Size of level labels
# 1, scalef - Scale of level labels (level value / scalef)
# Data Manipulation
# 100, column - # of columns for the input array (neg - flip vertical)
# -100, rownum - # of rows for the input array (neg - flip horizontal)
# 0, rotnum - number of 90 degree clockwise rotations
# Plot Axis Parameter
# 8.0, length of the x-axis
# 5.0, height of the y-axis
# 1.5, x-org
# 1.0, y-org
# Plot Parameters
# 0, xmin - min x axis label value
# 60, xmax - max x axis label value
# 2200., ymin - min y axis label value
# 0., ymax - max y axis label value
# 6, connum - number of contour levels (if 0 program will choose)
# 0, codesw - toggle for line weighting. must have 'connum' of codes.
# Contour levels if specified by user
# 75
# 80
# 85
# 90
# 95
# 100
# Codesw weighting values
# 1
# 2
# 0
# 1
# 2
# 0

```

EXAMPLE 3 : DEEP WATER (0,0,100,0) \*\* TILTED BEAM (2,8)





#### 11.4. Deep Water Example II (Range Independent)

This last example is of deep water propagation at 100 Hz in a range independent medium [1]. The source, 250 meter deep is a Gaussian source with a  $2^\circ$  width and a  $8^\circ$  tilt. The maximum water depth is 2000 *m* with a 100 *m* sediment layer and a subbottom. The script produces a plot of transmission loss versus range, a plot of transmission loss versus depth and a partial contour plot. One can see agreement with [1]. Script, input, plot control files and plots follow. This run takes 30 minutes on a Sun Worstation 3/50 (Sun Workstation is a registered trademark of Sun Microsystems, Inc).

Apr 20 08:50 1988 script Page 1

```
#
#
# EXAMPLE 4
#
pe << INN
example.4
INN
#
# part the data and plot the results
# of transmission loss versus range
#
part 281 133 outfil2sio temp
demux 1 temp temp1
multc .001 temp1 temp3
demux 2 temp temp2
mux temp3 temp2 temp
2n << INN > tlrde.plt3
tlranl.con
n
temp
temp
INN
#
# plotwin tlrde.plt3
#
# part the data and plot the results
# of transmission loss versus depth
#
part 1 90 outfil3sio ttemp
2n << INN > tlrde.plt3
trdep.con
n
ttemp
ttemp
INN
#
# plotwin tlrde.plt3
#
# script to treat the contour plot
#
part 67 33 outfil4 tempcont
getchan tempcont tempco 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
contour << INN > contourplot
pltconl
tempco
n
INN
plotwin contourplot
```

```

Feb  8 09:12 1988  getchan Page 1.

# get_chans - make a new sio data file from selected channels of an old
#               sio data file
if ($#argv < 3) then
  echo ' '; echo 'use: get_chans infile outfile chan1 chan2 ...';
  echo ' '; exit
endif
if (! -e $1 ) then
  echo ' '; echo "$1 not found."; echo ' '; exit
endif
set infile=$1 outfile=$2
shift; shift
echo getting channel $1 from $infile
demux $1 $infile $outfile
shift
while ($#argv >= 1)
  echo getting channel $1 from $infile
  demux $1 $infile $$
  mux $outfile $$ $outfile
  shift
end
rm $$

```

```

# input file for the program pe (MPL's pareq version)
# title of the input data file
Sacant Test #2 : Deep Water
# enter the 6 file names to save the output data
##### 7 characters is the required format
outfill1 # sectors information file
outfill2 # transmission loss versus range at selected depths
outfill3 # transmission loss versus depth at selected ranges
outfill4 # sio data file for contour
# sio field file
# tranform size (0 means that pe takes care of it)
0
# range step (0 means it is variable in pe)
0
# reference sound speed in m/s (can be 0, <0, >0)
0
# output range distance in m
100
# frequency in Hz
100
# source depth in m
250
# maximum range of track in km
45.
# operational mode of parabolic equation model (0 or 1)
0
# flag for spreading (0=cylindrical 1=spherical)
0
# source funtion (0-gaussian,1-modal,2-greene,3-thompson)
0
# half beam width, beam tilt from horizontal in degrees
1. 8.
# name of the sio data file for the field
# number of receiver depths, of selected ranges for tl vs range or depth
1
# number of ranges, number of depths for contour plot
100 100
# input receiver depth array for tl vs range in m
100.
# input selected range array for tl vs depth in m
40000.
# number of sectors ( integer)
1
# repeat for each sector :
# water depth in m, starting range in km
2000. 0
# number of points in the water sound velocity profile ( integer)
5
# sound velocity profile : depth in m, velocity in m/s
0. 1480.
100. 1476.
500 1470

```

```

1000. 1476.
2000. 1490.
# sediment heigth in m, sediment density in g/cm**3
100. 1.5
# attenuation in sediment in dB/wavelength
0.5
# number of points in the sediment sound velocity profile ( integer)
2
# sound speed profile in sediment : depth in m, velocity in m/s
0. 1475.
100. 1600.
# subbottom density in g/cm**3 and attenuation in dB/wavelength
2.0 0.1
# sound velocity in subbottom m/sec
1800.

```

Feb 8 09:02 1988 outfill Page 1

SOUND SPEED PROFILE IN WATER # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1480.00
100.00	1476.00
500.00	1470.00
1000.00	1476.00
2000.00	1490.00

SOUND SPEED PROFILE IN SEDIMENT # 1 AT RANGE 0.0 M

DEPTH(M)	SPEED(M/S)
0.00	1475.00
100.00	1600.00

SEDIMENT DENSITY = 1.500  
SEDIMENT ATTENUATION = 0.500

SUBBOTTOM DENSITY = 2.000  
SUBBOTTOM ATTENUATION = 0.100  
SUBBOTTOM SOUND SPEED = 1800.000

GLOBAL PLOT CONTROL PARAMETERS

numplt- number of data files plotted on one plot (2)  
 xlen- x axis length in inches (3)  
 noaxe - axis suppression. (0) suppress, (1) box, (2) lower, (3) mid (4)  
 nsf- sampling frequency flag (5)  
 nexpx- x axis exponential control (6)  
 # INDEPENDENT X AXIS PARAMETERS (8)  
 xorg- x axis origin in user units (9)  
 xinc- x axis sample increment in user units (11)  
 # AXIS SCALING PARAMETERS (12)  
 xmax- x maximum in user units (13)  
 xmin- x minimum in user units (14)  
 xtic- x tic mark spacing in user units (15)  
 numbers parallel (16)  
 size (17)  
 justify (19)  
 nxp- number of places after the decimal point (17)  
 # FILE PARAMETERS (17)  
 # filin- input d.a. file name (19)

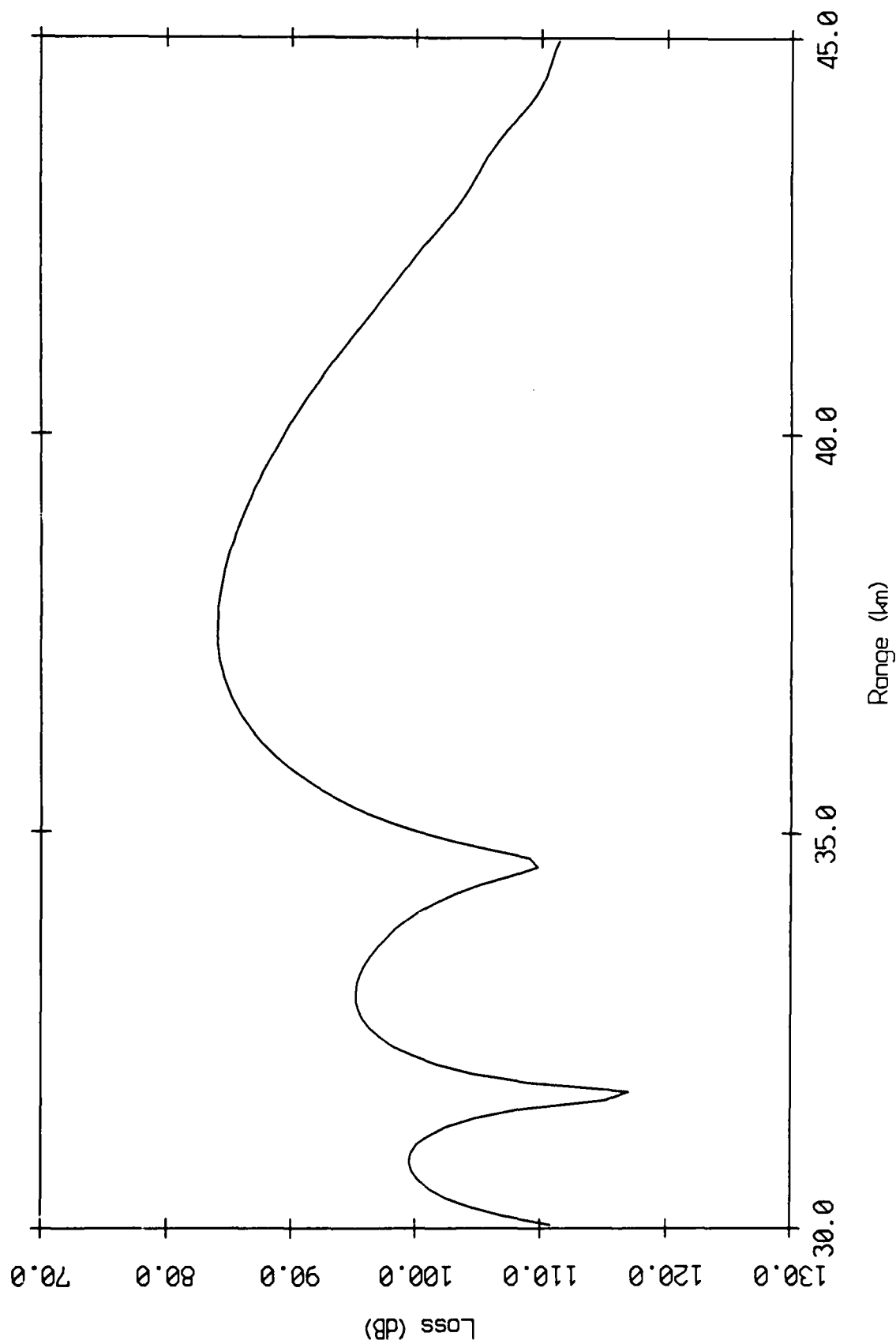
SETUP FOR INDIVIDUAL PLOT

x3 - lower left corner x position (2)  
 y3 - lower left corner y position (3)  
 ylen - length of axis (4)  
 ymax (5)  
 ymin (6)  
 ytic (7)  
 nrot (8)  
 size (9)  
 dec justify (10)  
 nyp - number of places after decimal place (11)  
 nlog - data manipulation control (12)  
 nexpy- y axis exponential control (13)  
 # YSTRNG - on axis (14)  
 Loss (dB) (16)  
 # desired MAJOR TITLE (18)  
 # desired MINOR TITLE (18)  
 Example 4 : Deep Water (0,0,100,0) \*\* TILTED BEAM  
 Freq.: 100 Hz, SD: 250 m, RD: 100 m, GAUSS (2,8)  
 0.20, titsiz - title size (20)  
 1.5, xloc - lower left x location (21)  
 6.7, yloc - lower left y location (22)  
 # desired Y TITLE (90 degrees to horizontal) (23)

0.8, xtloc - lower right x position (25)  
 2.0, ytloc - lower y position (26)  
 0.25, ytsiz- y title size (27)  
 # AXIS LABEL PARAMETERS (29)  
 # XSTRNG- x axis label string (29)  
 Range (km) (31)  
 2, chan1 - channel number to plot (31)  
 1, chan2 - channel to do math with (32)  
 -1, nreal- flag for plotting real or imaginary part-also da (33)  
 1, ndp0- first point plotted for partial file plotting (34)  
 0, ndtot- total number of points plotted for partial plot. (35)  
 1, nskip- plotting point increment (36)

numsyn- symbol number for point plotting (37)  
 symsiz- symbol size in inches (38)  
 xoff- x offset in user units (39)  
 0.0, yoff- y offset in user units (40)  
 0, ndash - number of dashed lines for this plot (41)

Example 4 : Deep Water (0, 0, 100, 0) \*\* TILTED BEAM  
Freq.: 100 Hz, SD: 250 m, RD: 100 m, GAUSS (2, 8)



GLOBAL PLOT CONTROL PARAMETERS

numplt- number of data files plotted on one plot (2)  
 xlen- x axis length in inches (3)  
 noaxe - axis suppression. (0) suppress, (1) box, (2) lower, (3) mid (4)  
 nsf- sampling frequency flag (5)  
 0, nexpx- x axis exponential control (6)  
 # INDEPENDENT X AXIS PARAMETERS (11)  
 xorg- x axis origin in user units (8)  
 1, xinc- x axis sample increment in user units (9)  
 # AXIS SCALING PARAMETERS (12)  
 60., xmax- x maximum in user units (13)  
 120., xmin- x minimum in user units (14)  
 5., xtic- x tic mark spacing in user units (15)  
 3, numbers parallel (16)  
 .125 size (17)  
 2, justify (18)  
 0, nxp- number of places after the decimal point (19)  
 # FILE PARAMETERS  
 # filin- input d.a. file name

DEPTH (M)

Example 4 : Deep Water (0,0,100,0) \*\* TILTED BEAM

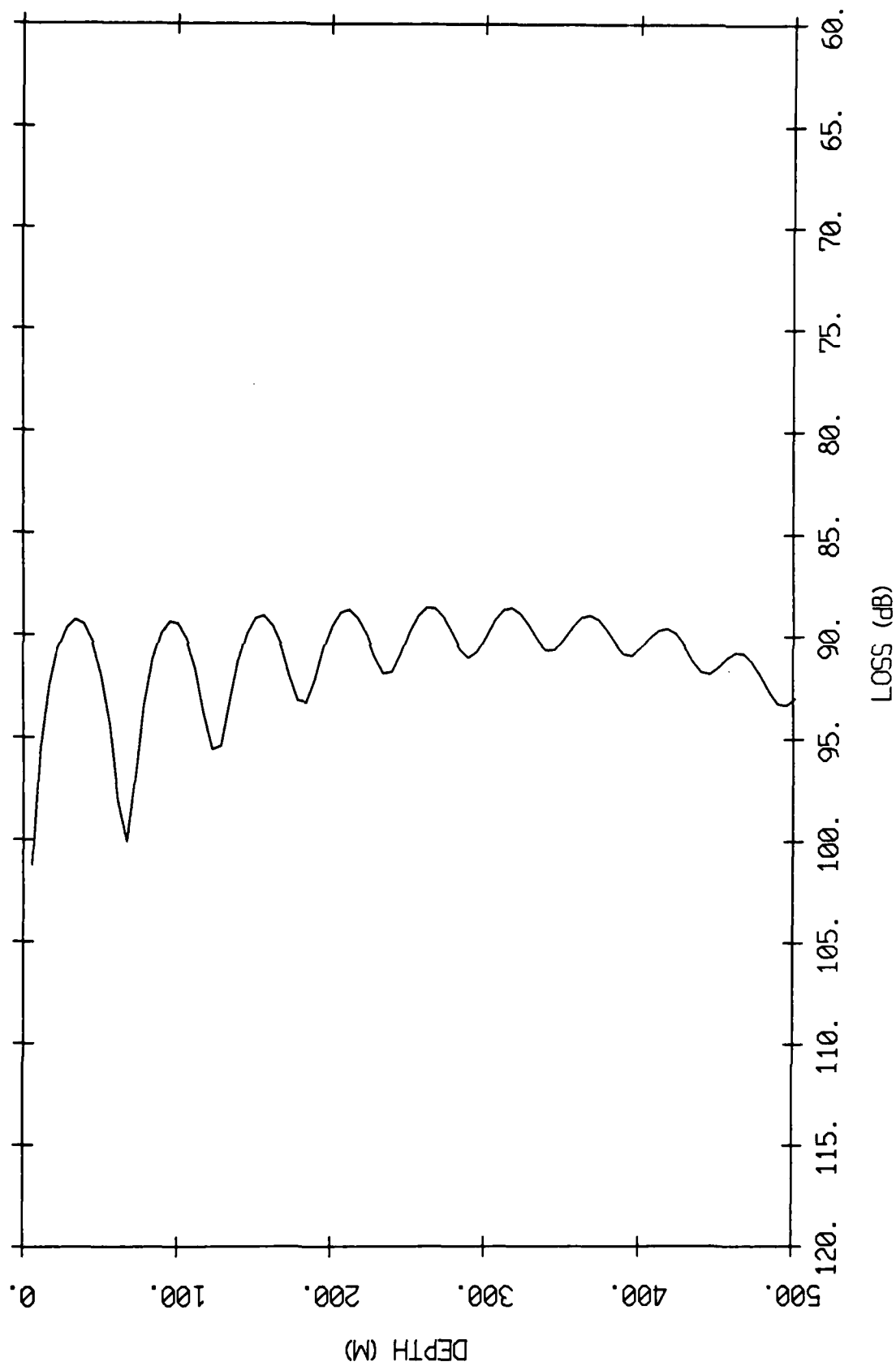
# desired MINOR TITLE (16)  
 # desired MAJOR TITLE (18)  
 Freq.: 100 Hz, SD: 250 m, RR: 40 km, GAUSS (2,8)  
 0.20, titsiz - title size (20)  
 1.5, xloc - lower left x location (21)  
 6.7, yloc - lower left y location (22)  
 # desired Y TITLE (90 degrees to horizontal) (23)

0.8, xtloc - lower right x position (25)  
 2.0, ytloc - lower y position (26)  
 0.25, ytsiz- y title size (27)  
 # AXIS LABEL PARAMETERS (29)  
 # XSTRNG- x axis label string (31)  
 1, chan1 - channel number to plot (32)  
 2, chan2 - channel to do math with (33)  
 -1, nreal- flag for plotting real or imaginary part-also da (34)  
 1, ndp0- first point plotted for partial file plotting (35)  
 0, ndtot- total number of points plotted for partial plot. (36)  
 1, nskip- plotting point increment

0, numsyn- symbol number for point plotting (37)  
 0.15, symsiz- symbol size in inches (38)  
 0.0, xoff- x offset in user units (39)  
 0.0, yoff- y offset in user units (40)  
 0, ndash - number of dashed lines for this plot (41)



Example 4 : Deep Water (0, 0, 100, 0) \*\* TILTED BEAM  
Freq.: 100 Hz, SD: 250 m, RR: 40 km, GAUSS (2, 8)



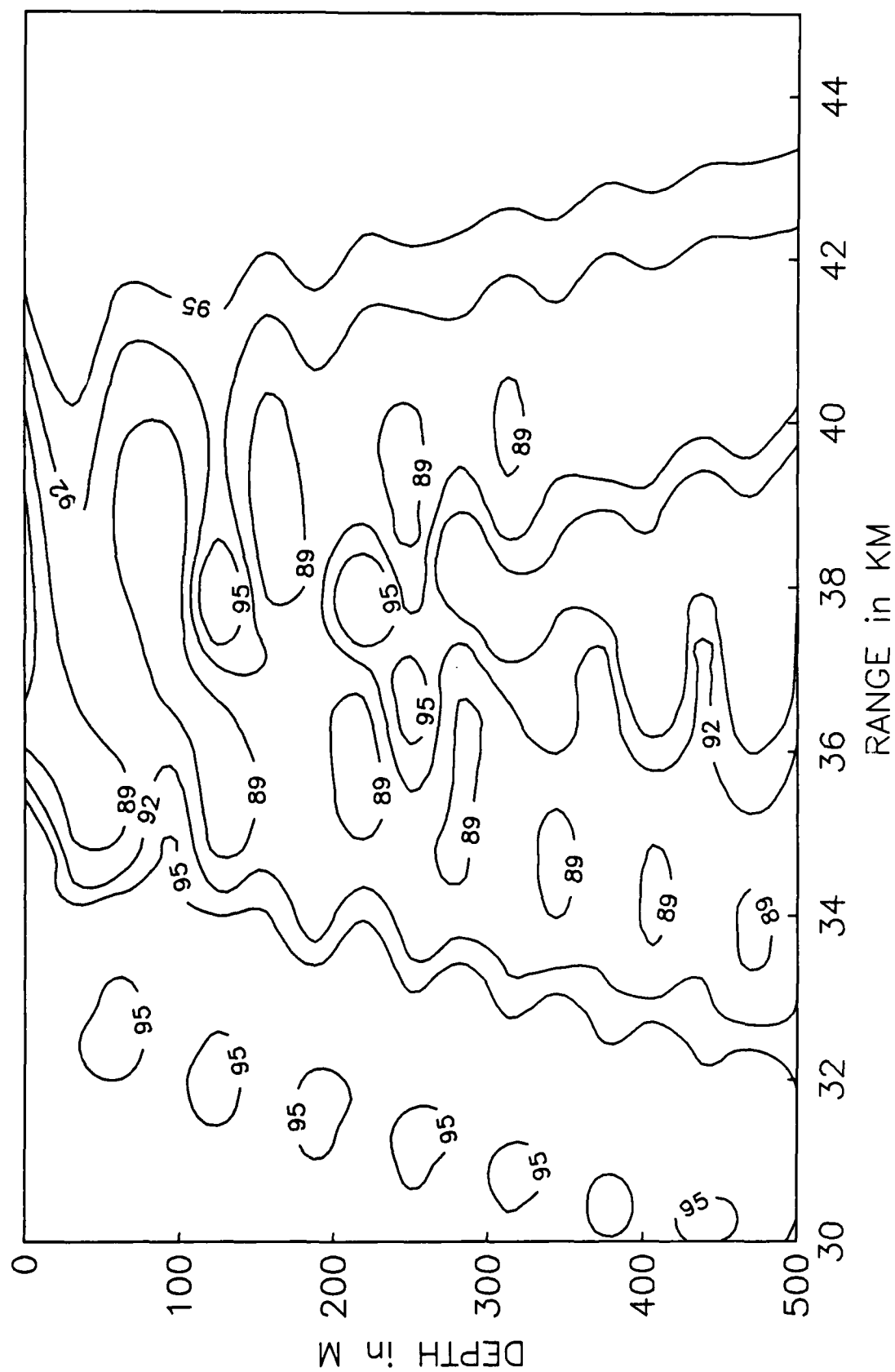
Apr 20 08:53 1988 pltconl Page 1

```

# Contour draft plot control file
# Labels
# title - label centered above plot
# caption - label at bottom left corner of plot ( left justified )
#
# RANGE in KM
# xlab - x axis label - centered at bottom of plot
# ylab - y axis label - centered at left side of plot
# DEPTH in M
# Label Sizes
# .15, titsiz - Size of Title
# .15, capsiz - Size of Caption
# .15, xlabsz - Size of x axis label
# .15, ylabsz - Size of y axis label
# .10, levsiz - Size of level labels
# 1, scalef - Scale of level labels (level value / scalef)
# Data Manipulation
# 33, colnum - # of columns for the input array (neg = flip vertical)
# -17, rownum - # of rows for the input array (neg = flip horizontal)
# 0, rotnum - number of 90 degree clockwise rotations
# Plot Axis Parameter
# 8.0, length of the x-axis
# 5.0, height of the y-axis
# 1.5, x-org
# 1.0, y-org
# Plot Parameters
# 30, xmin - min x axis label value
# 45, xmax - max x axis label value
# 500., ymin - min y axis label value
# 0., ymax - max y axis label value
# 6, connum - number of contour levels (if 0 program will choose)
# 0, codesw - toggle for line weighting. must have 'connum' of codes.
# Contour levels if specified by user
# Codesw weighting values
# 80
# 83
# 86
# 89
# 92
# 95
# 1
# 2
# 0
# 1
# 2
# 0

```

EXAMPLE 4 : DEEP WATER (0,0,100,0) \*\* TILTED BEAM (2,8)



## 12. References

- [1] F.B. Jensen, M.G. Martinelli, " The SACLANTCEN Parabolic Equation Model (PAREQ)", (VAX/FPS Version), December 1985.
- [2] F. Jensen H. Kroll, "The Use of the Parabolic Equation Method in Sound Propagation Modelling" SACLANTCEN Memorandum SM-72, 15 August 1975.
- [3] W.A. Kuperman, "Models of Sound Propagation in the Ocean", Naval Research Review 3 vol XXXVII, pp 32-41, 1985.
- [4] F. Jensen, "Acoustic Propagation Modelling, Why Several Models ?", SACLANTCEN CP-19, 1976.
- [5] F.D. Tappert, "Selected Applications of the Parabolic-Equation Method in Underwater Acoustics", International Workshop on Low Frequency Propagation and Noise, Vol I, Oct 14-19 1974, WoodsHole, Mass., Publ. 1977, pp 155-200.
- [6] F.D. Tappert, "The Parabolic Approximation Method", in Keller J.B. and Papadakis J.S. Eds, Wave Propagation and Underwater Acoustics, Lecture Notes in Physics 70. New York, Springer Verlag, 1977, pp 224-281.
- [7] D.J. Thomson, N.R. Chapman, "A Wide Angle Split Step Algorithm for the Parabolic Equation", JASA 74 (6), pp 1848-1854, December 1983.
- [8] M.D. Feit, J.A. Fleck Jr., "Light Propagation in the Graded-Index Optical Fiber", Applied Optics Vol 17 No 24, pp 3990-3998, 15 December 1978.
- [9] F.B. Jensen, H. Schmidt, "Review of Numerical Models in Underwater Acoustics Including Recently Developed Fast Field Program", SACLANTCEN Report SR-83, 15 December 1984.

- [10] H.K. Brock, "The AESD Parabolic Equation Model", NORDA Technical Note 12, January 1978.
- [11] S.T. MacDaniel, "Propagation of normal mode in the Parabolic Approximation", JASA 57 (2), pp 307-311, February 1975.
- [12] R.M. Fitzgerald, "Helmholtz Equation as an Initial Value Problem with Application to Acoustic Propagation", JASA 57 (4), pp 839-842, April 1975.
- [13] A.D. Pierce, "The Natural Reference Wavenumber for Parabolic Approximations in Ocean Acoustics", Comps & Maths with Appls, Vol 11 No 7/8, pp 831-841, 1985.
- [14] H.K. Brock, R.N. Buchal, C.W. Spofford, "Modifying the Sound Speed Profile to Improve the Accuracy of the Parabolic Equation Technique", JASA 62 (3), pp 543-552, September 1977.
- [15] L.E. Kinsler, A.R. Frey, Fundamentals of Acoustics, Wiley Eds
- [16] R.R. Greene, "The Rational Approximation to the Acoustic Wave Equation with Bottom Interaction", JASA 76 (6), pp 1764-1773, December 1984.
- [17] F.B. Jensen, W.A. Kuperman, "Sound Propagation in a Wedge-Shaped Ocean with a Penetrable Bottom", JASA 67 (5), pp 1564-1566, May 1980.